2001年8月1日発行（毎月1回1日発行）第20巻第8号 昭和57年12月6日第三種郵便物認可

マイコン
わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

AUG.2001 8



定価 550Yen

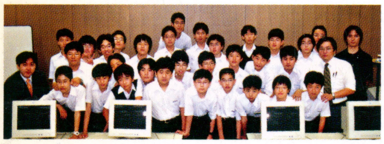


わかる！おもしろい！

プレステ2でGAMEプログラム！ 特集

《夏休み公開講座開催！》

本誌投稿を目標にみんなで学習
編集部の学校訪問記
神奈川県横浜市
サレジオ学院中学校・高等学校



エレクトロニック研究所訪問
海洋科学技術センター

電子工作入門
レゴ・マインドストーム
「センサーで自動ドアを作る」
GPSでロボットを自動操縦
「レゴ・ナビゲーション・システム」

好評連載 言語別プロテクコーナー

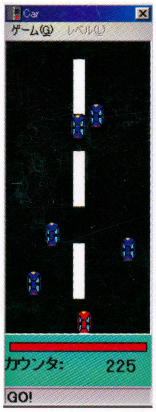
EXCEL VBA	「コントロールを使ってみる②」
DirectX	「サウンド処理で表現力を高める」
Visual Basic	「複数種類の敵キャラを登場させる」
ワンダーウィッチ	「インターネット接続に挑戦」
携帯電話	「iアプリで音楽再生」
JAVA	「プログラムを実行してみる」

読者が作った作品をドーンと紹介
特選投稿プログラムコーナー

VisualBasic6.0	THE カーチェイス
VisualBasic6.0	DESTRUCTION GOD
VisualBasic6.0	漢字でGO
VisualBasic6.0	（超大作）Logicer
VisualBasic6.0+DirectX7	（超大作）Pandora Bottle
Excel VBA	CUBEST
VisualC++V6.0	（移植版）Walls in the Box
F-BASIC V6.0	マウスde世界陸上
HSP V2.5	流れ☆
QBasic	聖騎士と魔王の塔
iアプリ	BLACK LIGHT
WonderWitch	ゼロタンク2
PC-E650	Fort Capture城塞攻略

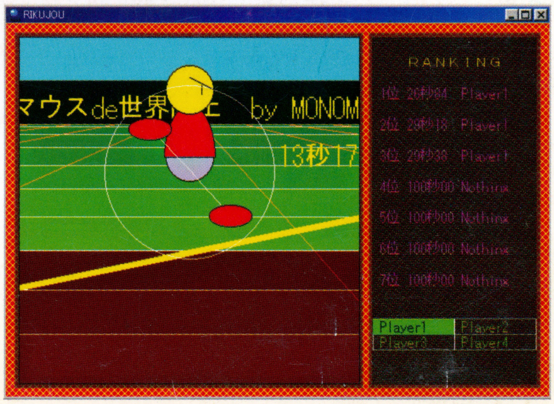
コンピュータ・ミュージック

SC-88ST Pro	南米の風
SC-88 Pro	Face to Face











専用ホームページ
http://www.basicmagazine.net/

アルファのフレッツ・ISDNコース 使い放題、毎日24時間使っても

月々わずか300円!!

NTT各種サービスも同時にお申込できます。

つなぎっぱなしの方には断然お得な専用コース!

フレッツ・ISDNコース

入会金無料!

月額300円(年額3,300円)

●常時接続 ●完全定額制(NTT料金別) ●HP容量15MB ●メールアドレス2つ無料(1つは弊社からの案内を含む受信専用アドレス)　※工事費1000円(初回のみ)

高速接続を重視される方にはぴったりの専用コース!

フレッツ・ADSLコース

ブロードバンド

入会金無料!

月額980円(年額10,780円)

●高速・大容量常時接続 ●完全定額制(NTT料金別) ●HP容量15MB ●メールアドレス2つ無料(1つは弊社からの案内を含む受信専用アドレス)　※工事費1000円(初回のみ)

いつでもどこでも使えるダイヤルアップの定番コース!

アルファ基本コース

モバイルもOK!

入会金無料!

月額980円(年額9,800円)

●無制限・使い放題(電話代別) ●HP容量15MB ●メールアドレス2つ無料(1つは弊社からの案内を含む受信専用アドレス) ●NTT、KDDI、各種携帯・PHS、モバイル使用可能

基本コースにプラス0円でISDNが使える充実コース!

アルファ基本コース + フレッツ・ISDNオプション

モバイルもOK!

入会金無料!

月額980円(年額9,800円)

●無制限・使い放題(電話代別) ●HP容量15MB ●メールアドレス2つ無料(1つは弊社からの案内を含む受信専用アドレス) ●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●常時接続、完全定額制(NTT料金別)　※工事費1000円(初回のみ)

自宅では高速接続、外出先からも使える万能コース!

アルファ基本コース + フレッツ・ADSLオプション

ブロードバンド

モバイルもOK!

入会金無料!

月額1,880円(年額20,680円)

●無制限・使い放題(電話代別) ●HP容量15MB ●メール アドレス2つ無料(1つは弊社からの案内を含む受信専用アドレス) ●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●高速・大容量常時接続、完全定額制(NTT料金別)　※工事費1000円(初回のみ)

※金額は全て税別です。

「フレッツ・ISDN」「フレッツ・ADSL」のアクセスポイント

全国47都道府県に拡大中

※一部市町村にご利用いただけない場合があります。

ダイヤルアップ全国268ヵ所アクセスポイント

詳しくはFAX情報サービス、Webサイトをご利用ください

(お申込はオンラインが簡単!)

24時間 オンライン申込受付中

www.alpha-net.ne.jp

24時間　資料請求受付電話(音声応答システム)

TEL:03-4306-0789

24時間　e-mail資料請求

info@alpha-net.ne.jp

24時間　FAX情報サービス(お申込書が入手できます)

TEL:03-3376-9898

[接続サービス一覧]

コース名称	アクセスポイント: 都道府県AP	全国共通AP	東京電話AP	携帯電話(au)AP	PHS(DDIポケット)AP	東京電話・中部アステルAP	「フレッツ・ISDN」	「フレッツ・ADSL」	HP: ホームページ容量(MB)	ホームページの商用利用	メールアドレス: 電子メールアドレス数	受信専用電子メールアドレス数	独自ドメイン電子メールアドレス数	ドメイン: 独自ドメイン
アルファ基本コース	○	○	○	○	○	○			15	×	1	1		
「フレッツ・ISDN」コース							○		15	×	1	1		
アルファ基本コース +「フレッツ・ISDN」オプション	○	○	○	○	○	○	○		15	×	1	1		
「フレッツ・ADSL」コース								○	15	×	1	1		
アルファ基本コース +「フレッツ・ADSL」オプション	○	○	○	○	○	○		○	15	×	1	1		
アルファ・ドメインコース	○	○	○	○	○	○			100	○			1	○
アルファ・ドメインコース +「フレッツ・ISDN」オプション	○	○	○	○	○	○	○		100	○			1	○
アルファ・ドメインコース +「フレッツ・ADSL」オプション	○	○	○	○	○	○		○	100	○			1	○

全国出張サポート受付中(有料)

アルファインターネットの接続から基本講習まで、全国に出張サポートいたします。詳しくはTEL.03-3377-6655までお電話ください。

●アルファ・ドメインコース以外のコースではホームページの商用利用は出来ません。●受信専用メールアドレスは後日Eメールにてお送りします。(アルファ・ドメインコースを除く)

あなたに、プラス

αinternet®

プロバイダ6年の実績が支える信頼のインターネットサービスを提供します。

つなぎ放題で独自ドメインが取れてホームページ容量100MBで月々わずか3580円!!

アルファに3タイプのドメインコース誕生
ブロードバンド対応、全国に拡大中!

アルファで得した分、おこづかい上げてよね。



New
月々3,580円でインターネット・カンパニー、ショップができます!

New
月々3,580円にプラス100円でISDNがご利用になれます!

New
月々3,580円にプラス200円でADSLがご利用になれます!

あなたの会社、ショップ名のドメインが取れるコース	さらにISDNが使えてリーズナブルな繁盛コース!	仕事に、趣味に、使えば納得の大充実コース!
アルファ・ドメインコース モバイルもOK! 入会金無料!	**アルファ・ドメインコース + フレッツ・ISDNオプション** モバイルもOK! 入会金無料!	**アルファ・ドメインコース + フレッツ・ADSLオプション** ブロードバンド モバイルもOK! 入会金無料!
月額3,580円(年額39,380円)	月額3,680円(年額40,480円)	月額3,780円(年額41,580円)
●独自ドメインの取得 ●HP容量100MB ●独自ドメイン電子メールアドレスが1つ ●ダイヤルアップ接続無制限・使い放題(電話代別)NTT、KDDI、各種携帯・PHS、モバイル使用可能	●独自ドメインの取得 ●HP容量100MB ●独自ドメイン電子メール1つ ●ダイヤルアップ接続無制限・使い放題(電話代別)●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●常時接続、完全定額制(NTT料金別) ※工事費1000円(初回のみ)	●独自ドメインの取得 ●HP容量100MB ●独自ドメイン電子メール1つ ●ダイヤルアップ接続無制限・使い放題(電話代別)●NTT、KDDI、各種携帯・PHS、モバイル使用可能 ●高速・大容量常時接続、完全定額制(NTT料金別) ※工事費1000円(初回のみ)

※金額は全て税別です。

営業時間
月曜～金曜 11:30～21:00
土曜11:30～17:00
日曜・祝日休み

アルファインターネット
〒151-0071 東京都渋谷区本町1-28-15
TEL:03-3377-6655 FAX:03-3376-7632
■通産省・電子商取引(EC)実証事業契約企業 ■電子商取引推進協議会会員

マイコン BASIC Magazine

CONTENTS

8
AUG. 2001

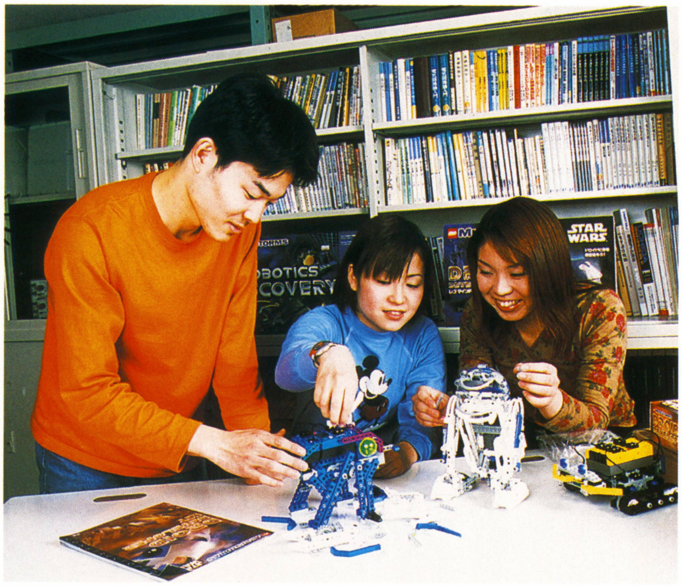
専用ホームページ http://www.basicmagazine.net/

広告さくいん

©マイコンBASICマガジン8月号
第20巻 第8号 2001年7月25日印刷
2001年8月1日発行
定価550円（本体524円）
電波新聞社　2001 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号141-8715
電話　東京(03)3445-6111（大代表）
振替　（東京）00150-3-51961
印刷：奥村印刷(株) 製本：(株)堅省堂

21世紀も日本電子が拓く。

1951年創立。今年で創立50周年を迎えます。
充実した施設・設備と多彩な学科構成で専門性の高いスペシャリストを育成しています。



一日体験入学

やったことなくたって、
見たことなくたって、
興味さえあれば大丈夫。

2001年開催日程

7月20日（祝）	8月 1 日（水）	8 月27日（月）
7月26日（木）	8月 2 日（木）	8 月29日（水）
7月27日（金）	8月23日（木）	9 月 9 日（日）
7月28日（土）	8月24日（金）	9 月23日（日）
	8月25日（土）	10月14日（日）

■全日ともAM10:00開催　事前申込み制　お申込みはフリーダイヤルまたはホームページで

おもちゃの製作とプログラミング

最新おもちゃの構造を知る
レゴ社にて開発された“レゴ・マインドストーム”（レゴブロックを使用したコンピュータ制御によるロボット）を使い、その仕組みやデザインなどについて初歩から体験します。最新のおもちゃの仕組みの裏側にせまります。

ロボットの制御プログラミング

最新ロボットの仕組みと制御技術
ロボットを動かすために、頭脳となるコンピュータにプログラムを書き込んで、目や手足となるセンサーやモータを制御します。また、動物型ロボットや仮想動物などの最新の制御技術を実演やビデオを通して説明します。

Windowsプログラミング入門

プログラムに挑戦
Visual Basicというプログラム言語を使って、“ハイパーGOGO双六（スゴロク）”を制作します。こちらで開発した「お助けツール」を使って作っていきますので双六（スゴロク）のルールさえ知っていれば初心者から経験者までOKです。

システム開発の基本技術

オリジナルゲームの制作

Linux C言語によるプログラミング

オリジナルCG画像の制作

などなど　やりたいことがきっと見つかる**全29コース**（資料をお送りします）



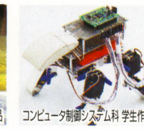
2002年度 新設

おもちゃ工学科

おもちゃ工学関連で日本初の学科が誕生。今、注目のペットロボットなどのハイテクおもちゃを創造するクリエイターを育成します。


コンピュータ制御システム科 学生作品


コンピュータグラフィックス研究科 学生作品


コンピュータ制御システム科 学生作品


ゲーム制作研究科 学生作品

ELECTRONICS系

おもちゃ工学科 ☆	2年
電子情報技術科	2年
電子応用工学科	2年
電気工学科 ※	2年
電気工事技術科 ※	2年

GAME系

ゲーム制作科 ※	2年
ゲーム制作研究科	3年
ゲームCGデザイン科	2年

IT系

コンピュータネットワーク科 ※	2年
コンピュータネットワーク研究科 ☆	3年
情報処理科 ※	2年
情報システム開発科	2年
シニア情報処理科	3年
コンピュータ総合技術科	2年
コンピュータ制御システム科	2年
インターネットビジネス科	2年
コンピュータビジネス科	2年
ITスペシャリスト科 ☆※	1年

CG系

コンピュータグラフィックス科 ※	2 年
コンピュータグラフィックス研究科	3 年
CG映像科	2 年
映像制作科	2 年
アニメーション科	2 年
コンピュータデザイン科	2 年

MUSIC系

ミュージックアーティスト科 ☆	2 年
ライブ・イベント科 ☆	2 年
ゲームミュージック科 ☆	2 年
コンピュータミュージック科	2 年
音響芸術科	2 年

☆は2002年4月新設学科　※は夜間部あり



Anniversary 50th e-ism
Make a Big Step 2001電子学園

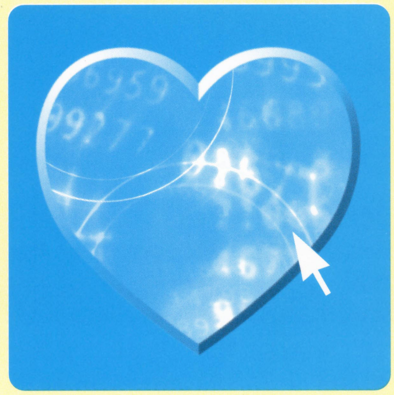
学校法人　電子学園

日本電子専門学校

〒169-8522　東京都新宿区百人町1-25-4　Tel.03（3363）2985　交通：JR総武線大久保駅南口徒歩2分、新宿駅西口徒歩10分
http://www.jec.ac.jp　iモード http://www.jec.ac.jp/i　9691　E-mail info@cons.jec.ac.jp　入学案内書無料送付　学校見学・入学相談随時受付

お問合せ・資料請求は　0120-00-9691（入学相談室）

自分発、新コミュニケーション



HOMEPAGE CONTEST

ホームページで伝える私らしさ、あなたらしさ。

第4回 ICF ホームページコンテスト

キミのことを、キミの言葉で伝えられるのは、世界中でキミだけなんだ。
それが出来たとき、キミからの新コミュニケーション

テーマ：「自分発、新コミュニケーション」

目的：自己表現を通じてインターネット時代のコミュニケーションを考える

募集期間：2001年7月16日（月）～9月30日（日）

応募資格：中学生、高校生

応募方法：ICFホームページコンテスト登録ページにURLでご応募ください。（http://www.icf.or.jp/contest/）

審査基準：独自性、質・量、コミュニケーションへの意志、双方向性、国際性、見やすさ・扱いやすさ等を総合的に評価します。

審査委員：高橋　徹（インターネット戦略研究所 会長）
林　徳治（山口大学教育学部 教授）
今泉　洋（武蔵野美術大学デザイン情報学科 教授）
太田　亨（ICF理事長）
阿久津　敏幸（KCOM代表取締役社長）ほか各氏

審査発表：2001年10月19日（金）ホームページ上で発表いたします。

表彰：［個人の部］最優秀賞（1作品）優秀賞（2作品）特別賞（1作品）
［団体の部］最優秀賞（1作品）優秀賞（2作品）特別賞（1作品）
入賞者にはコンピュータ周辺機器を贈呈いたします。
さらに副賞として専門学校等に於いてプロの指導者によるホームページ制作授業が受講できます。

表彰式：2001年11月10日（第2土曜日）午後
KDDIホテルストラーダ（東京都新宿区）

お問合せ：E-mail:contest@icf.or.jp　Fax:03-5202-8509

主催：財団法人　国際コミュニケーション基金（ICF）

後援：総務省
電波新聞社　KDDI株式会社　株式会社KCOM

KDDI　KCOM KDDI Group

お問合せ　E-mail:contest@icf.or.jp　Fax:03-5202-8509

エレクトロニック研究所訪問

海洋科学技術センター

http://www.jamstec.go.jp/　　レポート：松原拓也





海洋科学技術センターは海洋の総合的開発利用を目的として1971年（昭和46年）に設立されました。研究施設は横須賀本部，横浜研究所，むつ研究所と3か所に点在しますが，今回はその中心となっている神奈川県横須賀市の横須賀本部を見学してきました。

本部には，深海の水圧を再現する高圧実験水槽装置や，潜水実験を行うための潜水シミュレータ装置などの研究設備，そして数多くの調査船や探査機などを有しています。

海洋観測

海洋科学技術センターでは，トライトンブイと呼ばれる観測機器を展開したり，「かいよう」などの調査船で海洋観測を行い，地球温暖化やエル・ニーニョ現象の解明のための研究をしています。

なお，2001年末には「地球シミュレータ（宇宙開発事業団・日本原子力研究所との共同）」のスーパーコンピュータによるシミュレーションも行われる予定です。

海洋底の観測

地球表面には何枚かのプレートが存在し（10枚という説など，研究者によって数はまちまち），その活動（プレートテクトニクスやマントルダイナミクス）が地震や噴火の原因となっています。



▲海洋調査船「かいよう」。全長61.5m，総トン数3，176トン。水面面積が小さく，波浪に強い半没水型双胴船。船首と船尾に計8台あるスラスタで旋回や横移動も容易

海洋科学技術センターでは，有人潜水調査船「しんかい6500」や，10,000m級無人探査機「かいこう」などの最新設備で海洋底を観測・研究しています。

この他，ナホトカ号の重油流出事故の調査や，打ち上げ失敗で海底に沈んだ「H-Ⅱロケット8号機」の第1段ロケット・エンジンの調査（1999年11月～2000年1月までの3回）でも，これらの設備が生かされています。

深海地球ドリリング計画

地震発生のメカニズム，火山噴火活動等の原因となる地球内部の構造を解明するために「深海地球ドリリング計画」が進められています。この計画では「地球深部探査船」を建造し，水深約2，500m（最終目標約4,000m）の海底から約7,000mを掘削する，ライザー掘削という最新技術が投入されます。



▲有人潜水調査船「しんかい6500」（1989年建造）。全長 9.5m。最大潜航深度は6，500m。建造は三菱重工業



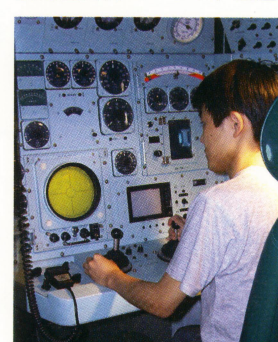
▶「しんかい6500」には2台のTVカメラと、圧力計、温度計などの測定機、そして7自由度のマニピュレータが2台装備されている。

海洋の生態系の研究

海底では，600気圧の海底で生息でき



▲深海調査研究船「かいれい」（1997年建造）。全長105.2m。総トン数4,628トン。航海速力約16ノット。建造は川崎重工



◀海洋科学技術館に展示されている「しんかい2000」の操縦室。実際は内径2.2mの耐圧殻に3名が乗り込む



▲「かいれい」のブリッジ奥にある，調査指揮・計算機室。ここから無人探査機「かいこう」の操縦を行う

る耐圧性細菌や，石油を分解する石油分解細菌，そして，光合成のできない場所で硫化水素やメタンをエネルギー源とする化学合成生物群集などの発見があり，これらは海洋生態・環境研究部と極限環境生物フロンティア研究システムで研究されています。

海洋エネルギーの利用

海洋観測以外にも，海洋環境の改善や海洋エネルギーの利用が研究されています。

たとえば，三重県五ヶ所湾に建造された沖合浮体式波力装置「マイティーホエール」は，波の力で電気を作り出す技術などを研究しています。さらに，静岡県焼津市に整備された「深層水分析研究棟」では，海底にある栄養塩を多く含んだ深層水の利用技術を研究しています。



▲「かいれい」の船内にはシリコングラフィクス製ワークステーションが並んでいた。調査した海底を計算により3Dモデルとして認識することが可能



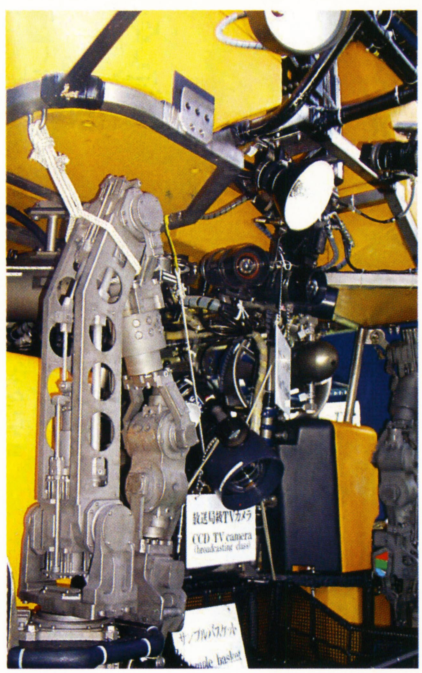
▲10,000m級無人探査機「かいこう」（1995年建造）。最大潜航深度は11,000m 。海中でランチャーとビークルの2機に分離する

横須賀本部の主な研究組織

深海研究部
海洋技術研究部
海洋観測研究部
海洋生態・環境研究部
極限環境生物フロンティア研究システム
固体地球統合フロンティア研究システム
深海地球ドリリング計画推進室

エル・ニーニョ現象とは

数年に一度，赤道太平洋の海面水温が広い範囲にわたって数度上昇する現象。世界の異常気象との関連性が研究されているが，その具体的なメカニズムはまだ解明されていない。



▲「かいこう」のビークルには4種類（全6台）のカメラと，照明灯，航海装置，7自由度のマニピュレータが2台搭載されている



▲高圧実験水槽装置では，深度6,500m相当の圧力を1,500回かけた後の圧壊試験において，深度13,200m相当の圧力で破壊されたチタン製の球体が展示されていた



▲深海総合研究棟にある深海微生物実験システム。深海の泥を環境状態を保持した状態で採取し，一度も地上の条件にもどすことなく，深海環境下で実験できる施設



▲無人探査機「ハイパードルフィン」。全長3.0m，最大潜航深度3,000m。通常のTVカメラより画質の良いハイビジョンカメラを搭載



▲「深海地球ドリリング計画」で建造中の「地球深部探査船」（写真は模型）。掘削能力は海底下7,000m。船名は7月中旬に発表予定



▲総水量約1000トンの潜水訓練プール。潜水訓練や潜水機器・水中機器の開発や試験に使われている



▲液晶パネルを多用して整然としている「ハイパードルフィン」の操縦コンテナ

▼パソコン部の生徒たち



ベーマガへの投稿を目標にしたパソコン部と，少人数制で学習するパソコン授業

編集部の学校訪問記

横浜市 サレジオ学院中学校・高等学校

http://salesio.netty.ne.jp/

横浜市中心部から北北西に約12km，東京都心部から南西に約25kmに位置する港北ニュータウン。豊かな自然環境に恵まれたその一角に，カトリック・ミッション・スクール，サレジオ学院（河合恒夫学校長）があります。中・高一貫教育により，コミュニケーションを大切に，バランスのとれた教育を実践している男子校です。



◀「バランスの取れた生徒を育成する」と教育方針を語る豊岡稔教頭先生





▲▶校舎の外観と，サレジオ会創立者の聖ドン・ボスコの碑

新校舎落成と同時にできたパソコン室

サレジオ学院が現在の横浜市都筑区に移転したのは平成7年。新校舎は，生物・化学・物理などの理科教室や，パソコン室・音楽・美術・技術家庭などの専門教室の充実を目指して設計されました。

パソコン室には，授業で使用するIBMのMMX200MHzパソコンが24台，そのほか，教室後方にNEC VALUESTAR V200をはじめとしたパソコン部用のパソコンが21台設置されています。

パソコン授業 少人数制で，きめ細かな指導が進む

高校1年生の「家庭一般」の授業を拝見しました。指導しているのは，技術科の江刺宏恭先生です。

ざっと見回してびっくりしたのが，授業を受けている生徒数が22人と少なめな点。これは，一クラス約40人を半分にわけ，一方ではパソコン学習，もう一方では，食物や被服を学習しています。テクニック面を重視する授業は，個々に先生の目が行きわたり，きめ細かな指導ができるようにと，一クラスを2班にわけているのだそうです。

家庭一般の授業ではホームページ制作が！

授業では，ホームページ作成を行っていました。テーマは自分のプロフィールページを作ることです。

「まずは，一方的にしゃべってしまいますので，わかる人はどんどん進めてください。わからない人は先生が各人を回ったところで

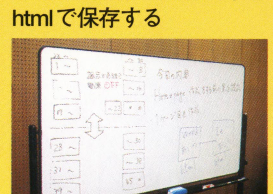


▲家庭一般の授業では「ホームページ」制作が行われていた



▲パソコン室にはIBMのMMX Pentium200マシンが

▼その日の授業の流れが書かれた白板。Word文章をhtmlで保存する





▲技術科の江刺先生。熱血派

パソコン部の目標はベーマガへの投稿！



▲取材時にはOBも訪れていて，後輩を指導していた



▲中学1年生にプログラムを指導している先輩。アットホームな雰囲気





◀▲パソコン部所有のパソコンは，部費で買ったものがほとんど。中1の1，2学期でN88-BASICを，3学期にVisual Basicを習得し，中学2年からは自分でそれぞれの目標に向かってプログラミングを行っている



▲パソコン部の浜田先生（左）と教育実習生の山田先生（右）を囲んで



▲部長の上符くん



▶元部長の山城庸平くん（高3）

質問してください」と，江刺先生は，まずその日にやるべきことをひととおり話してしまいます。

作成方法は，Wordで作った文書をhtmlで保存し，Internet Explorerで，保存したhtmlファイルを開いてみるという方法です。

さらに「Internet Explorerの[表示]→[ソース]で，ホームページを構成しているHTMLが見られます。このタグをいじれる人はいじってもOKです」と，各人の力量によって授業は進められます。「理解できない生徒には個別に時間もかけられますし，少人数制の良さを生かしています」(江刺先生)。

その後の授業では，お互いに出来上がったホームページを見せ合い，メールのやりとりをしていく方針だそうです。

パソコン部 目標は「ベーマガに投稿すること」

パソコン部は，8年前に同好会として発足，3年前に「部」に昇格しました。中学1年から高校3年までが一緒に活動しており活気に満ちあふれています。現在では，33人が所属しています。

内容は，N88-BASICやVisual Basicでのプログラミング，Klick＆Play，RPGツクールなどでのゲーム制作が主で，基本的に週3回活動しています。

目標は，毎年秋に行われる「サレジオ祭」での作品展示と，各種コンテストへの参加（神奈川新聞と岩崎学園主催の「中高生ソフト・アイデアコンテスト」で特別賞受賞の実績がある）。そして，最大目標は，ベーマガへの投稿だそうです。

現部長の上符浩太郎くん（高1）は，2000年2月号にN88-BASIC用「授業」が掲載されました。またパソコン部を指導している浜田明巳先生（数学科）も2001年1月号にVisual Basic用「五心Simulation」で採用されています。OBも過去には何人か採用されたそうです。

また感心したのは，入部したばかりの中学1年生に，先輩たちがプログラムの組みかたを親切丁寧に教えていること。入ってきたばかりの生徒のほとんどがプログラムを組むこと自体初めてなのに，1年でプログラムが組めるようになるのだそうで，今後が楽しみです。

ホームページ



▲沿革やカリキュラムなど，しっかりできたページ



▲IPIXというプラグイン・ソフトを使って校舎を360度で見ることができる

サレジオ学院中学校・高等学校 〒224-0029 横浜市都筑区南山田3-43-1

わかる！おもしろい！初歩からのプログラミング講座

プレステ2でゲームをプログラムする！



入門編

「BASIC STUDIO」アシスタントプロデューサー ●アートディンク 齋藤博希

夏休み公開 プログラミング講座開催！

プログラムの楽しさをみなさんに体験していただく「BASIC STUDIO」を使用した“夏休み公開プログラミング講座”の開催が7月30日（月）にきまりました（詳細は19ページをご覧ください）。そこで，本誌ではこの公開講座に先がけて，講座に参加する人たちの事前予習の意味と，出席できない人たちのために，本講座の内容をできるだけわかりやすく紹介したいと思います。

すでに本誌で連載されている「BASIC STUDIO パワフルゲーム工房入門」の内容と重複する個所もありますが，復習を兼ねて読んでいただければと思います。

プログラムするために何が必要？

さて，BASIC STUDIOを始めるにあたっていくつか用意するものがあります。必要なものは第1表のとおりです。

この中でUSBキーボードとマウスはPlayStation 2本体の前面にあるUSB端子に接続して使用します。これらの機器がなくてもひととおりの操作はできるようになっていますが，あったほうが格段に使いやすくなります。

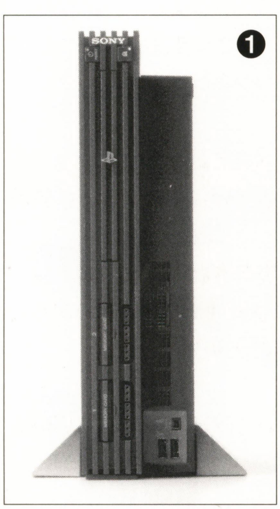
キーボードを使うのが初めてという人は使い慣れていないために戸惑うこともあるかと思います。しかし，コントローラによる入力と同じくらいの入力スピードであれば，キーボードのほうが使い慣れていない分，慣れたときの入力スピードは格段にちがいます。はじめは入力する文字数も多くないので，練習の意味でもキーボードを購入して使うことをお勧めします。

そうそう，表には書かれていませんが，いちばん必要なのはあなたの「やる気」かもしれませんね。

ポリゴン表示も可能なBASICだ

最近のゲーム機の進歩はすばらしく，私がはじめてゲームに触れたときに比べると格段にきれいなゲーム画面が表示されるようになりました。逆に読者のみなさんにとってはこのようなきれいな画面のゲームが当たり前の世界になっています。

ゲームはグラフィックがすべてというわけではありませんが，やはり作るなら「きれいな画面で作りたい」と思うのは自然な心理だと思います。



◀PlayStation2本体。最新モデルはSCPH-30000，35,000円（6月29日から改定された）

《第1表》BASIC STUDIOを使うのに必要なもの

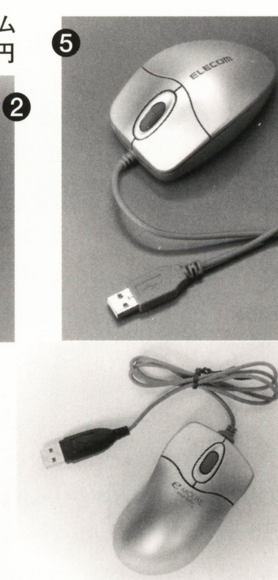
PlayStation 2 本体	①
BASIC STUDIO	②
メモリーカード	③
USB キーボード	④
USB マウス	⑤

▼「BASIC STUDIO パワフルゲーム工房」パッケージと内容物。9,800円

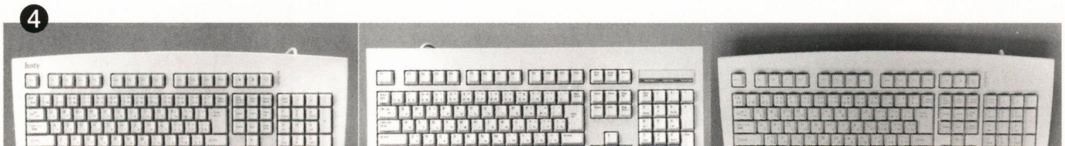




◀プレイステーション2専用メモリーカード（8MB），3,500円。PlayStation2本体（SCPH-30000）には付いていない



▲USBマウス。上がエレコムの「M-WUCSV」（2,980円）。下がサンワサプライの「MA-EMUS」（3,800円）



▲USBキーボード。左がジャスティの「UKB-112B」（オープン価格＝アートディンクの通信販売で3,480円）。中がエレコムの「TK-U109 JPW2」（5,000円）。右がサンワサプライの「SKB-112UH」（6,900円）

BASIC STUDIOでは，プログラムを行う部分は簡単なBASICを採用していますが，PlayStation 2の能力を使って簡単にポリゴンモデルなどの表示ができるようになっています。

また，画面分割や残像などの個性的な命令も用意されているので，きっとあなたの作りたいゲームが作れると思います。

なお，ゲームを構成するポリゴンモデルや2Dグラフィック，BGMなどを作成するツールはBASIC STUDIOで用意されています。サンプルの素材も多数用意されているので，「モデルは作れるけど曲は作れないよ」といった人でもサンプルから好きな物を選んでくることができるようになっています。

それではプログラミングを始めよう

それでは，EZ-BASICを使ってプログラミングを行いましょう。

EZ-BASICを立ち上げると，

```
EZ-BASIC Version 1.0
```

とだけ表示され，画面には「カーソル」と呼ばれる白い四角形が点滅しているだけで，このまま見ているだけで何かがはじまるといったことはありません。このように，こちらが何もしなければ，BASICも何もしてくれません。

それでは，キーボードを使って，つぎの文を入力してみましょう。

```
PRINT 100 [Enter]
```

［Enter］はキーボード右側のEnterキーを押すということです。また，「PRINT」と「100」の間はスペースをあけます。すると，画面にはつぎのように表示されます。

```
100
OK
```

この「PRINT」とは後に書かれた文字を画面に表示させてくれるものだということがわかるでしょう。つまりここではPRINTの後に入力された「100」が表示されたのですね。

このようにBASICに入力することで，BASICが仕事をしてくれるものを「命令」と呼びます。まさにあなたが「これやって」とコンピュータ（ここではPS2）に命令するわけです。

命令には，さまざまな種類があり，PRINTのように画面上に文字を表示し，目にみえる反応をするものもあれば，BASIC内部に働きかけるタイプの命令もあります。命令の数は多いですが，簡単な英単語をもとにしているので，少しずつ，覚えていきましょう。

命令の書きかた

私たちが普段使っている言葉にも文法があるように，BASICの命令にも文法といったものが存在します。とはいっても複雑に考える必要はありません。先ほどの入力したものを思いだしてください。はじめに「PRINT」といった仕事をさせる命令があり，つぎに「100」といった表示させるものが書かれています。「100」に当たる部分は「パラメータ」と呼ばれ，命令を補助する内容をここに入力します。

たとえばあなたが「持ってきて」とだけ言われても何を持ってくるのかわかりませんね。「タオルを持ってきて」のように「タオル」といったパラメータがあってはじめて「命令」を実行することができるのです。

このようにBASICでは基本的に

「命令コマンド」「パラメータ」

（※命令コマンドとパラメータの間はスペースを空ける）

という書きかたでBASICに命令を実行させるようになっています。

このような書きかたの規則を「構文」と呼び，構文にしたがって書かれた命令を「命令文」と呼びます。なお，パラメータの書きかたや内容は命令によってちがってきます。

入力間違いを知らせる“エラー・メッセージ

BASICでは命令文に間違いがあった場合，命令を実行するときに親切に知らせてくれます。たとえば，つぎのように命令文をわざと間違って入力してみます。

```
PRIN 100[Enter]
```

すると先ほどとちがって，

```
PRIN 100
構文の誤りです。
```

と表示されます。このように間違った構文を入力すると，BASICが間違いを指摘してメッセージを表示します。このメッセージを「エラー・メッセージ」と呼びます。逆にこのエラー・メッセージを読むことで，自分の書いた命令の間違いがわかるようになっています。

たとえば，「構文が間違っています」というエラー・メッセージは，

「命令」「パラメータ」

という構文の約束が守られていないことを示しています。つまり，「PRIN」という命令がBASICにはないため，構文としての約束が守られていないと判断したのです。

陥りやすい間違い

初心者が主に起こす間違いとして，下記のようなものがあげられます。

- 命令文のスペルミス
  PRINT（正）→ PLINT（誤）など

- 似たように表示される文字
  1（数字のイチ），I（英大文字のアイ），l（英小文字のエル）など

これらの間違いに注意して入力を行いましょう。

文字を表示しよう

つぎはPRINT文を使って「HELLO」と表示させてみることにします。

```
PRINT "HELLO"[Enter]
```

《第２表》 プログラムでの四則演算に使う記号

＋	足し算	＊	掛け算
－	引き算	／	割り算

さて，気づかれたかと思いますが，なぜHELLOの文字の両端を「"（ダブルクォーテーション）」で囲んだのでしょう？

これは後ほど説明する「数値変数」と呼ばれるものと区別をするために「"」で囲む必要があったのです。試しに先ほどの入力例から「"」をはずして表示すると実行結果は「0」と表示されてしまいます。入力した文字そのもの（これを文字列と呼びます）を表示したい場合は両端に「"」をつけることを覚えておいてください。

数値と文字列のちがいは？

これまでのPRINT文で使用したパラメータを見てわかるとおり，パラメータとして扱うことのできるものは，数字を表す「数値」と文字そのものを表す「文字列」の二つに大きくわけることができます。

数値は「数字」の集まりで，計算を行ったり，どちらが大きいかなどの比較を行うことができます。命令のパラメータの大半がこの数値形式になっています。

文字列は「数字」や「英文字」，「日本語」「記号」の集まりの両端を「"（ダブルクォーテーション）」で囲んだものです。主にメッセージなど文字の表示に使います。

計算をさせてみよう

BASICで扱う数値は通常の数と同じように計算を行うことができます。今度は，重要な項目でもある数値の計算を行ってみましょう。まずは，つぎのように入力してください。

```
PRINT 1+2[Enter]
```

入力を行うと，計算結果である「3」が表示されたはずです。このように計算結果を表示するにはPRINT文のパラメータに計算させたい式を記入するだけで行うことができます。

これはBASICでは「命令のパラメータとして入力された数式は計算された値が使われる」というルールがあるためです。よって，先ほどの命令ではPRINT命令の後に書かれた「1＋2」は計算結果の「3」が表示されたのです。

逆に「1＋2」をそのまま表示したい場合は，どうしたらいいのでしょう？そうです，先ほどの「HELLO」と同じように「1+2」を「"」で囲んで「"1+2"」と入力すればよいのです。

なお，式に使う四則演算の記号は**第２表**のようになっています。

また，文字列は，基本的に四則演算をすることはできませんが，例外的に足し算をすることができます。

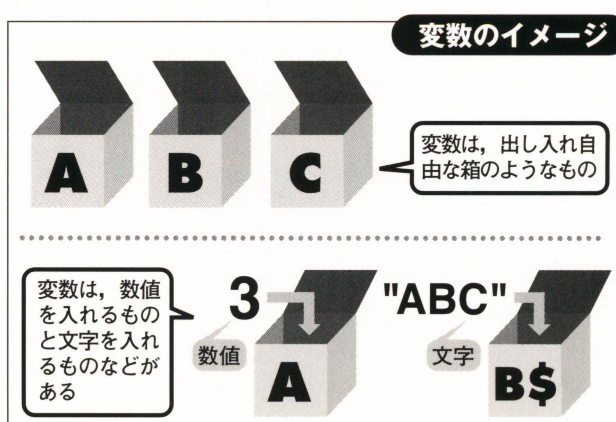
```
PRINT "AB"+"CD"[Enter]
```

どうでしょう?「AB」と「CD」が足されて「ABCD」と表示されたはずです。

変数を使ってみよう

先ほどのBASICに計算をさせる項目では，計算したい数値を直接パラメータに記入していました。しかし，4+5のような同じ足し算を行いたい場合，再度PRINT 4+5といった命令を入力する必要があります。今は単なる足し算なので再入力の手間はかかりませんがこれが複雑な計算式になると大変です。

このようなときは変数と呼ばれるものを使うと同じ計算式を使いまわすことができます。変数とは数値や文字列を入れておくことのできる「入れ物」のようなもので，先ほどの足し算でパラメータに数値ではなく，使う「入れ物」を指定してあげれば，BASICでは指定した「入れ物」に入っている数値を見て計算をしてくれます（**第１図**）。



《第１図》 変数は，数値や文字列を一時記憶させる場所

変数に数値を入れてみよう

変数は入れ物のようなもの，と説明しましたが，逆に変数は単なる入れ物なので中身を入れなければ意味がありません。変数を使った計算式を使う前に変数に数値を入れてみましょう。

```
LET A=10[Enter]
```

「LET」は命令，パラメータの「A=10」の「A」は変数の名前，「10」は変数に入れる数値です。「LET」命令は，パラメータとして入力された「＝」の左側に書かれた変数名(ここでは「A」)に右側の「10」という数値（数式でも構いません）を代入する命令です。

しかし，この命令文を実行しても「OK」と表示されるだけで画面上の変化はありません。そこで「A」に「10」が代入されていることを確認するために，つぎの命令文を実行してみましょう。

```
PRINT A[Enter]
```

PRINT文によって，「A」という変数を表示させると「10」が表示され，「A」に「10」が代入されていることがわかりました。

なお，パラメータの「A=10」は算数で使う等式に似ていますが，入力した＝の左側の変数に右側の数値を代入するといった意味ですので，算数の等式とはちがうことに注意してください。左側が数値で右側が変数名の①や左側が数式となっている②はエラー・メッセージが出て実

行されません。逆に，③のように右側の「A＋1」という数値を新たに左側の「A」に代入することで，変数Aの値に1を加えるといった使いかたをすることができます。

また，命令コマンドの中でLET命令だけは命令コマンドを省略して，④のように入力することができます。

```
LET 10=A ・・・・・・・・・・・①
LET A/2=10 ・・・・・・・・・②
LET A=A+1 ・・・・・・・・・・③
B=30 ・・・・・・・・・・・・・④
```

それでは先ほどの計算を変数を使って行ってみましょう。変数を使うと，つぎのようになります。

```
A=4[Enter]
B=5[Enter]
PRINT A+B[Enter]
```

変数を使うことで3回命令を入力する手間がかかっていますが，同じ計算をちがう数値で計算をしたい場合は，変数A，Bに入れる数値を変更するだけでできます。このようにする利点は計算式で使われる要素の管理が行いやすくなることです。また，変数は何度でも使うことができるので，計算結果を別の変数に代入して他の部分で使うことなどもできます。

文字列にも変数がある

数値に変数があるように，文字列にも変数が用意されています。文字列変数は数値変数と区別をされるために変数名に$がつき，代入したい文字列は「"(ダブルクォーテーション)」で囲みます。

```
A$="HELLO"[Enter]
B$="How"+"are"+"You"[Enter]
```

このようにして文字列変数への文字列の代入を行います。基本的には数値変数の場合と同じですね。

変数を使った足し算プログラム

変数に数値を代入して計算を行うときのように，複数の命令文を使って処理を行いたいことがよくあります。このようなときに，いちいち命令文を一つ一つ実行していたのでは，やはり手間がかかります。

このようなときは，命令文の前に命令文を実行する順番を記した「行番号」を描くことで，複数の命令文をまとめた「プログラム」にすることができます。たとえば変数を使った足し算のプログラムは，つぎのようになります。

```
10 A=4[Enter]
20 B=5[Enter]
30 PRINT A+B[Enter]
```

このように入力すると，各命令文の最後で［Enter］キーを押してもすぐには実行されず，複数の命令文をひとまとめにした「プログラム」として，EZ-BASICに登録されます。

登録された「プログラム」を実行するにはRUN［Enter］と入力します。

```
RUN[Enter]
```

すると，先ほどと同じように4+5の答えである9が表示されます。

なお，今まで使ってきた行番号を使わないで行う方法は「ダイレクト・モード」と呼びます。

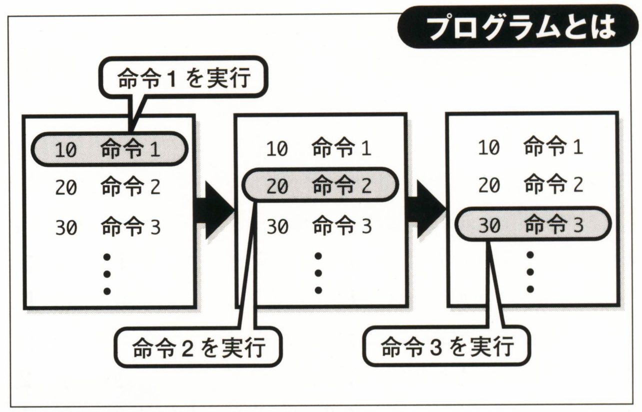
プログラムの流れ

プログラムは基本的に行番号の順に実行されるので，プログラムを書いた順番には影響されません。たとえば先ほどのプログラムにさらに命令文を追加した後，RUNで実行してみましょう。

```
25 A=A+10[Enter]
```

今度は25行にAの値に10加えるという命令が加わったため，30行で行う足し算は14+5となります。よって，このプログラムを実行すると19と表示されます。

このようにプログラムの流れを確認するには行番号の順番を追うことが重要になってきます（第2図）。



《第2図》プログラムは命令を順番に並べたもので，一つずつ順番に実行される

プログラムを表示するLIST命令

何度かプログラムを実行していると，実行結果の表示などで自分が入力したプログラムが画面から消えてしまいます。入力したプログラムを再度表示して確認するには，「LIST」命令を使います。

```
LIST[Enter]
```

すると，すでに登録済みのプログラムが表示されます。このようにLIST命令はプログラムを行番号の順にはじめから最後まで表示します。

また，LIST命令と同じような働きをするものとして，「VIEW」命令があります。「VIEW」命令を実行するとLIST命令と同様に入力されたプログラムが表示されますが，1画面を超えたプログラムはカーソル・キーで上下にスクロールさせることができます。入力したプログラムが長い場合はこちらのほうを使うとよいでしょう。

プログラムの修正，追加

プログラムを書いていると命令文の追加や削除をしたり，ミスをした部分の修正を行うことがあります。

ここではプログラムの編集について説明します。

命令文の修正

一度プログラムとして登録した命令文を修正するには，LIST命令などで修正したいプログラムを表示した後，修正したい部分を書き直して［Enter］キーを押せば，命令文の再登録が行われます。［Enter］キーを押さないと，修正した命令文が登録されませんので注意しましょう。

命令文の追加

命令文の追加は先ほど説明したとおり，追加したい部分のあいている行番号を使って命令を入力します。

もし，何度も追加をして空いている行番号がない場合は，後に説明する「RENUMBER」を使いましょう。

命令文の削除

使わなくなった命令文など，必要のなくなったプログラムを消去するには，行番号のみを書いて［Enter］キーを入力してみましょう。先ほどの修正の応用で不要な行を消去することができます。

はじめからプログラムを書き直す

入力したプログラムをすべて消去して，新しくプログラムを書き直したいときは，「NEW」命令を使用します。

```
NEW[Enter]
```

「NEW」命令を使用すると，今までに登録したプログラムをすべて消去します。消したプログラムは元に戻せないので，使うときは十分に注意が必要です。

だからといって以前のプログラムがメモリ上に残ったまま，新しいプログラムを入力すると新しく入力したプログラムと昔のプログラムが混在して実行しても動作がおかしくなってしまいます。このような事態を避けるためにも，新しいプログラムを入力する際は「NEW」命令を実行しましょう。

行番号を整理する

プログラムの追加や削除を行っていると行番号の増えかたにバラツキができてしまい，プログラムが見にくくなったり，命令文を追加するための隙間がなくなってしまいます。そのような場合はダイレクトモードで

```
RENUMBER
```

を実行すると，登録されているプログラムの行番号を自動的に10刻みに書き直してくれます。

INPUT文を使ってみる

変数を使った計算をしてみたものの，先ほどのプログラムはプログラムを書いている段階で変数へ数値を入力しているため，あまり使い勝手がよくありません。

そこで，プログラムを書いている段階ではなく，実行している段階で変数への数値の代入を行える「INPUT」命令を使ってみることにします。一度「NEW」を実行してから，つぎのプログラムを入力してみてください

```
10 INPUT A
20 INPUT B
30 PRINT A+B
```

INPUT命令はパラメータに書かれた変数にキーボードで入力された値を代入する命令です。それでは試しにこのプログラムを実行させてみましょう。

「実行を開始します。」が表示された後に「？」が表示されます。これは10行を実行してBASICが変数Aに代入する数値を，あなたに聞いてきているのです。変数Aに代入する数値を入力するには，キーボードから数値を入力して[Enter]キーを押してください。

さらに「？」が表示されるはずです。今度は変数Bに代入する数値を入力します。すると，先ほど入力した二つの数値を足した値が30行のPRINT文によって表示されます。

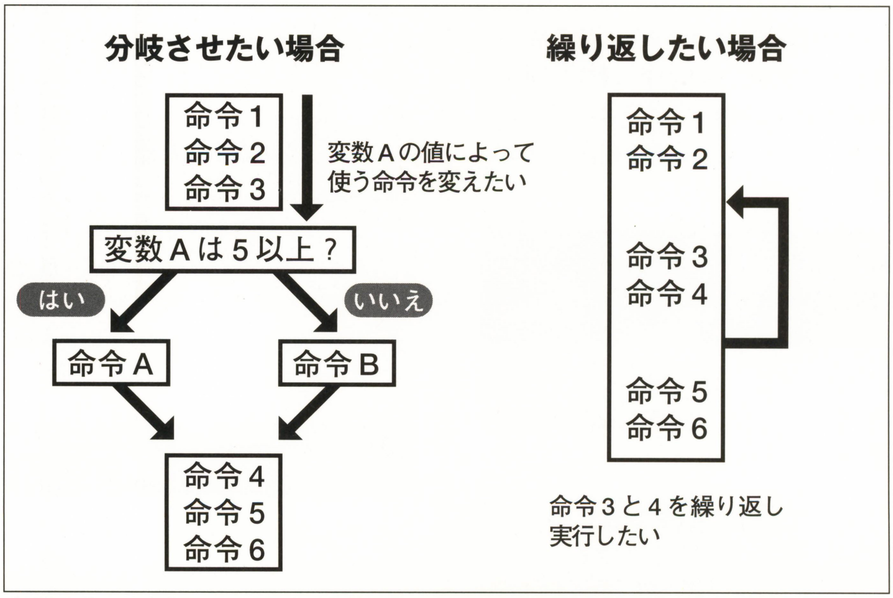
プログラムの流れを制御する

プログラムが実行される流れは先ほど説明したとおり，行番号の若い順に実行されます。しかし，時には同じ命令文を繰り返し実行させたり，条件によって実行させる命令文を変えたいときもあるでしょう（第３図）。

BASICの命令の中にはこのようなプログラムの実行の流れを変化させるような命令もあります。

条件によって実行する命令文を変更する

まずは設定された条件によって，実行する命令を変更するための「IF ～



《第３図》 プログラムの流れは，条件によって分岐させたり，繰り返させたりできる

THEN」命令について解説しましょう。
「IF～THEN」は「もし～ならば～を実行する」といった意味の命令で，「IF」と「THEN」の間に書かれた条件を満たしていれば「THEN」の後ろに書かれた命令を実行します。
ここでは，例としてINPUT文で入力した数値が5未満であれば「SMALL」とメッセージを表示するプログラムを作ってみることにします。

```
10 INPUT A
20 IF A<5 THEN PRINT "SMALL"
30 PRINT "END"
```

このプログラムでは20行で「IF」と「THEN」の間にある「A<5」が「Aの値が5未満である」という条件を表しています。
たとえば，10行で入力されたAの値が3だった場合は条件を表す「A<5」が「3<5」となり，「この式は正しい」つまり条件を満たしていると判断されます。条件を満たしている場合はTHEN以降の命令を実行するので，「SMALL」と表示され，つぎの行に移ります。
逆にAの値が9だった場合は「9<5」となり，「この式は間違っている」＝条件を満たしていないと判断され，THEN以降の命令を実行せずにつぎの行に移るのです。

論理式と比較演算子

IF～THEN命令で条件として書かれている「A<5」は論理式と呼ばれます。論理式では，変数や数値を比較してその結果が正しいか，正しくないかを判定します。
なお，二つの値を比較する「<」を比較演算子と呼びます。比較演算子の見かたは第３表のようになります。

《第３表》 比較演算子 「<」 の意味

＝	左側と右側が同じかどうか比較
＞	左側が右側より大きいか比較
＜	左側が右側より小さいか比較
＞＝	左側が右側以上かどうか比較
＜＝	左側が右側以下かどうか比較
＜＞ ＞＜	左側と右側が違っているか比較

IF文で実行される命令を複数設定する

先ほどのプログラムの書きかただと，「THEN」命令の後に実行できる命令は一つだけとなります。条件を満たしている場合の命令を二つ以上設定したい場合には20行の「THEN」のあとに何も書かず，つぎの行から実行させたい命令を書きます。この場合，1行で終わるIF文とちがって，実行させたい最後の命令文を書いたらつぎの行に「END IF」命令を入力し，「IF」文で使う命令の終わりを示す必要があります。

```
10 INPUT A
20 IF A>5 THEN
30  PRINT A
40  PRINT "SMALL"
50 END IF
60 PRINT "END"
```

条件を満たさない場合にも命令を実行する

「ELSE」命令を使うとIF文で条件を満たさなかった場合に命令を実行させることができます。条件を満たさなかった場合に「ELSE」命令の後に入力された命令文が実行されます。
なお，IF文を閉める「END IF」は「ELSE」実行される命令文の後に入れてください。

```
10 INPUT A
20 IF A<5 THEN
30  PRINT "SMALL"
40 ELSE
50  PRINT "BIG"
60 END IF
70 PRINT "END"
```

このプログラムを実行した場合はINPUT文で入力したAの値が5未満であれば「SMALL」を，5以上であれば「BIG」を表示します。
このような条件による分岐は，「右ボタンが押されていれば，右に移動する」や「左ボタンが押されていれば，左に移動する」といったように，さまざまな場所で使います。

同じ命令を繰り返す

つぎに，繰り返しに関する命令を説明しましょう。繰り返しもさまざまなところで使うことができ，大きなところでは“タイトル画面→ゲーム・スタート→ゲーム・オーバー→タイトル画面”といった繰り返しのほか，細かいところではプレイヤー・キャラクターを少しずつ右にずらして表示することを繰り返せばキャラクターの移動を表現することができます。

DO～LOOPを使ったプログラムの繰り返し

たとえば1から100までを表示するプログラムを作る場合，つぎのようにいちいちPRINT文を書いていくのは非常に手間のかかることです。

```
10 PRINT 1
20 PRINT 2
30 PRINT 3
40 PRINT 4
 ・
 ・
```

ここで，DO命令とLOOP命令を使います。DO，LOOP命令は，LOOP命令まで処理の流れがきたら，対応するDOに戻ることで，二つの命令の間に挟まれた命令文を繰り返し実行させる命令です。

```
10 A=1
20 DO
30  PRINT A
40  A=A+1
50 LOOP
```

このプログラムでは50行のLOOPに処理が来た時点で，対応するDO，つまり20行のDOまで戻って処理を続行します。これにより，30，40行の処理が繰り返し行われることになります。

プログラムの強制中断

先ほどのプログラムを実行するとすごい勢いで数が増加して画面の上に流れて

いってしまいます。プログラムを見て流れを追うとわかるとおり，50行まで命令文が実行されると20行に戻ってしまい，いつまでも終わらないプログラムとなっています。この状態を「無限ループ」と呼びます。

とりあえずこのプログラムを中断するにはコントローラのSELECTボタンとSTARTボタンを同時に押してください。また，USBキーボードを接続している場合は「Pause Break」キーを押してください。これで現在実行しているプログラムを中断することができます。

終了条件の設定

このような無限ループを回避するには終了条件を設定します。それでは変数Aの値が100になったら終了するようにしてみましょう。

終了条件を設定するには先ほどのプログラムのLOOP命令の行をつぎのように変更します。

```
10 A=1
20 DO
30  PRINT A
40  A=A+1
50 LOOP UNTIL A>100
```

「UNTIL」は「UNTIL」の後に書かれた条件を満たす(この場合は変数Aが100を超えた場合) までは「LOOP」を実行するというものです。

プログラムの実行速度を制御する

先ほどのプログラムの流れは1から100までの数字を順番に表示しているのですが，実際にプログラムを実行すると1から100までの数字が一度に表示されているように見えます。これはプログラムの実行速度が非常に速いために，一度に表示したように見えるのです。

数字を順番に表示されているのがわかるようにするには「FSYNC」命令を使います。「FSYNC」命令は最後に実行された「FSYNC」命令から数えて，1秒間に60回行われているテレビの画面書き換えを入力されたパラメータの数だけ待つ命令です。

たとえば先ほどのプログラムをこのように修正すると1秒ごとに数値が増えるプログラムとなります。

```
10 A=1
20 DO
30  PRINT A
40  A=A+1
45  FSYNC 60
50 LOOP UNTIL A>100
```

また，「FSYNC 1」の命令文を同じようにメイン・ループに入れると，テレビの画面を書き換えるタイミングと同期が取れるので，表示がちらつかないという利点があります。必ず入れるようにしましょう。

制御系命令のまとめ

制御系の命令のまとめとして文字が画面上を左から右に移動するようなプログラムを組んでみましょう。

まず，PRINT命令の働きをおさらいします。今までPRINT 命令を「文字（数値)」を表示するものとして使ってきましたが，表示される文字の位置はどこだったでしょう？今まで使ったPRINT 文の結果を思いだすと，PRINT文を実行した際のカーソルの位置に表示されるということがわかるはずです。

ということは，文字を右に移動させるには「カーソルの位置を右にずらしてから文字を表示すればよい」ということになります。ここで使うのが「カーソルの位置を指定する」LOCATE 命令です。

LOCATE命令でのカーソル位置を指定するにはテキスト座標を使って行います。テキスト座標とは，画面左上を基準として，左から右に伸びているx軸，上から下に伸びているy軸を使って，「x軸の方向に何文字分，y軸の方向に何文字分」といった形で表す「座標」です（第4図)。

それでは実際にLOCATE 命令の後にPRINT 命令を実行してみましょう。

```
10 LOCATE 18,11
20 PRINT "A"
```

どうでしょう，画面真ん中あたりに「A」と表示されたと思います。このようにLOCATE命令を使えば，好きな位置に文字を表示することができます。

これを利用して文字が右に移動するプログラムを作成してみましょう。

```
10 X=0
20 DO
30  LOCATE X, 11
40  PRINT "O"
50  X=X+1
60  FSYNC 10
70 LOOP UNTIL x>=37
```

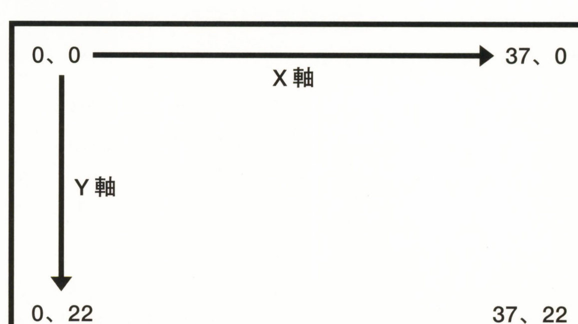
このプログラムではXの値を連続的に変化させることで30行で指定する座標を変化させ，40行で表示する文字の位置を右にずらしています。

しかし，このままですと，先ほどまで書いたプログラムや，PRINT文によって表示された「O」が画面上に残ってしまいます。

画面上のテキスト文字を消去する

ここで画面上の文字を消す命令「CLEAR」を使用します。「CLEAR」命令は，［Esc］キーを押したのと同様に画面上の文字をすべて消します。ただし，カーソルの位置がテキスト座標の原点(0，0)に移動してしまうので，LOCATE命令を使うときには実行させる順番に注意が必要です。

「CLEAR」命令を使うと，つぎのようなプログラムとなります。



《第４図》 画面の座標

```
10 X=0
20 DO
30  CLEAR
40  LOCATE X,11
50  PRINT "O"
60  X=X+1
70  FSYNC 10
80  LOOP UNTIL x>=37
```

40～50行で新しい座標に「O」を表示する前に，30行で画面上のテキスト文字を消去して，古い座標の「O」を消しています。これで「O」が右に移動しているような表現をすることができました。

この「文字を右に移動させる」といった動作を

1. 古い座標の文字を消す
2. 新しい座標を指定
3. 新しい座標に文字を表示

といった一連の流れ（これをアルゴリズムといいます）によって表現されていることを覚えておいてください。

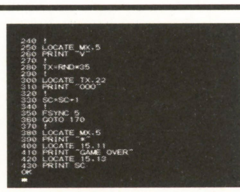
ではまた来月号で

さて，今回の講座はここまでです。今回登場した命令を第4表にまとめてみました。次回はここまでに使った命令＋αを使って簡単なゲームを作成します。お楽しみに！

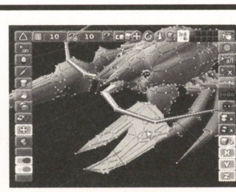
《第4表》 今回の講座で使用した命令一覧（登場順）

今回使った命令	基本的な書式	意味
PRINT	PRINT 数値または文字列	パラメータに書かれた数値，または文字列を表示します。
LET	LET 変数＝数値，または文字列	変数に値を代入する
RUN	RUN	プログラムを実行する
LIST	LIST	プログラムを表示する
VIEW	VIEW	プログラムを表示する（カーソル・キーで移動可能）
NEW	NEW	登録されたプログラムを破棄する
RENUMBER	RENUMBER	プログラムの行番号を10刻みにする
INPUT	INPUT 変数	キーボードから入力された数値（文字列）を変数に代入する
IF～THEN	IF 条件を示す論理式 THEN 命令文	条件を満たした場合，THEN 以降の命令文を実行する
END IF	END IF	IF 文で指定する命令文の終わりを示す
ELSE	ELSE	IF 文で指定した条件を満たさない場合，ELSE 以降の命令を実行する
DO	DO	繰り返しはじめ（必ずLOOPとセットで使う）
LOOP	LOOP	繰り返し終わり（必ずDOとセットで使う）
FSYNC	FSYNC 画面書き換えを待つ回数	画面の書き換えを指定回数待ってから次の命令に進む
LOCATE	LOCATE テキストX座標、テキストY座標	カーソル位置を指定した座標に移動する
CLEAR	CLEAR	画面に表示されたテキスト文字をすべて消去する

BASIC STUDIOを構成する五つのツール



EZ-BASIC
BASICでプログラミングする



POLY CRAFT
3Dポリゴンのキャラクターを作成する

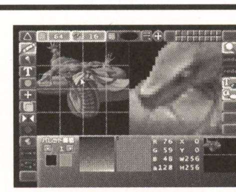
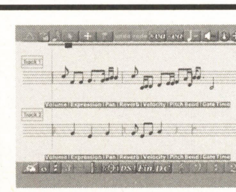
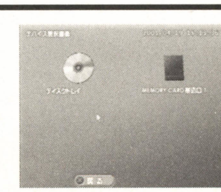


IMAGE-SKETCH
2Dグラフィックやポリゴンのテクスチャを描く



SCORE-NOTE
楽譜の形で音楽を作る

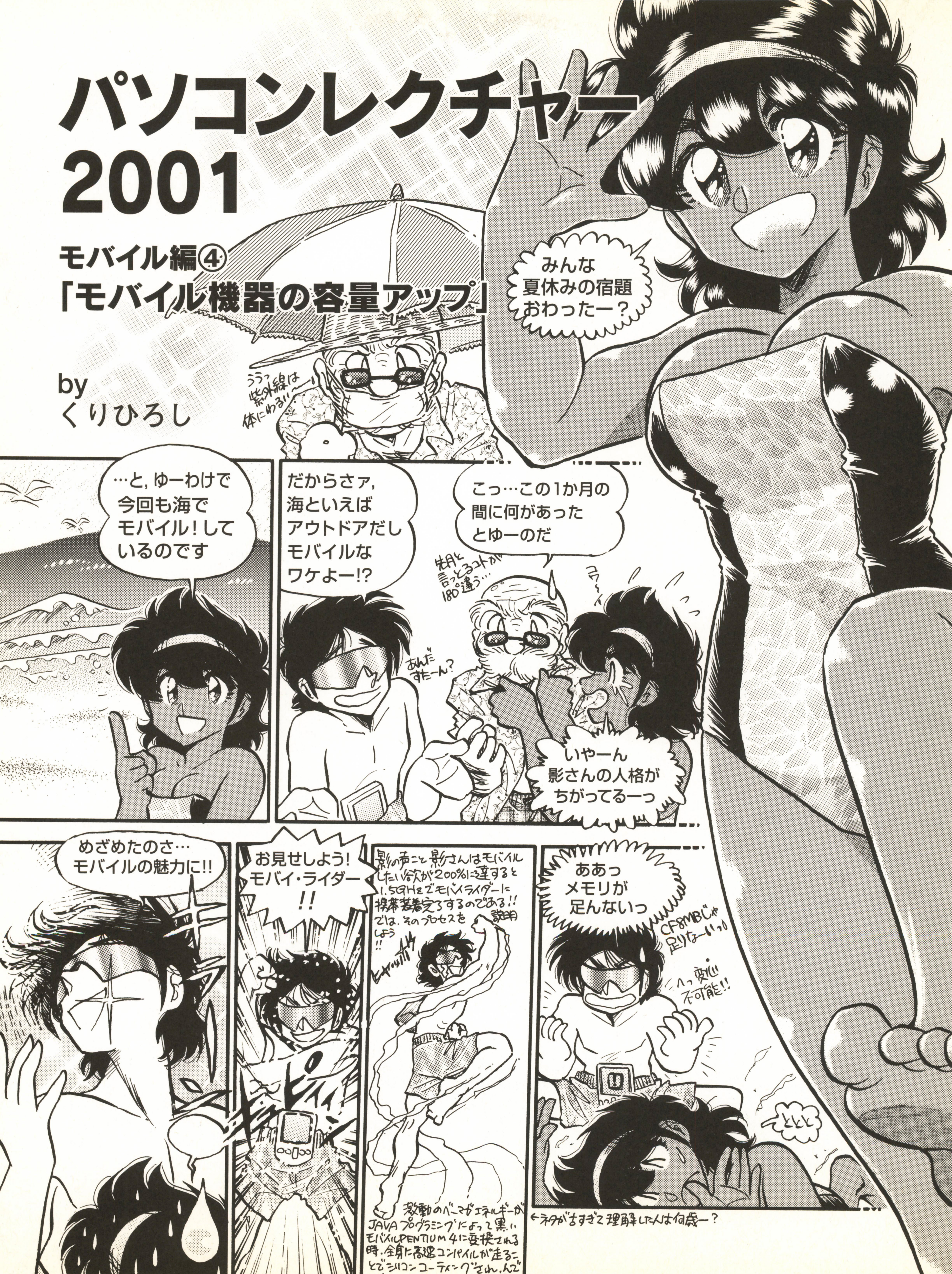


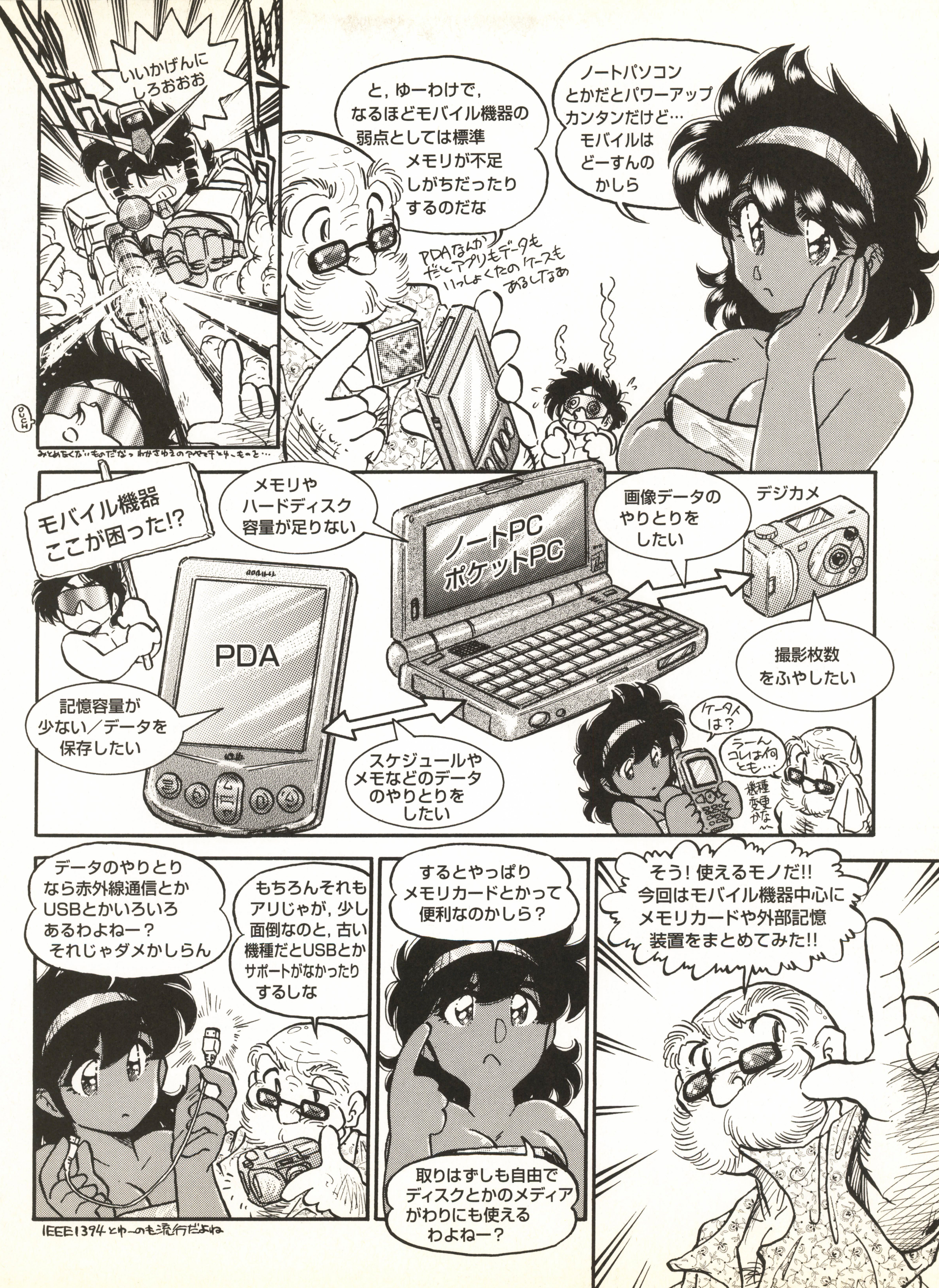
B-SHELL
ファイル管理やプログラムの実行をする

パソコンレクチャー 2001

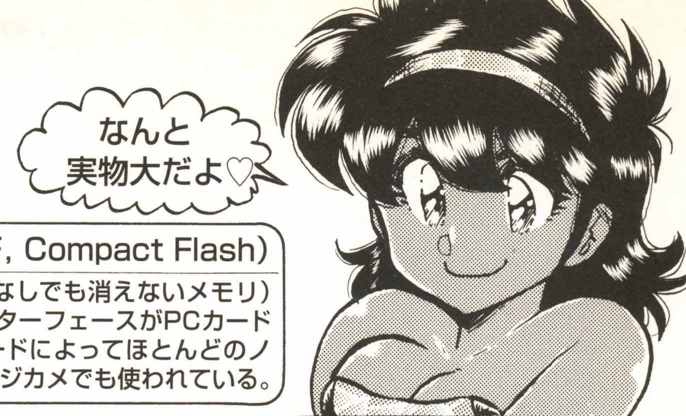
モバイル編④「モバイル機器の容量アップ」

by くりひろし

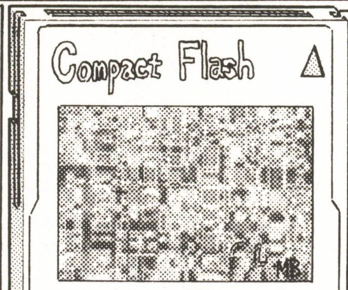



ノートパソコンとかだとパワーアップカンタンだけど…モバイルはどーすんのかしら
と，ゆーわけで，なるほどモバイル機器の弱点としては標準メモリが不足しがちだったりするのだな
PDAなんかだとアプリもデータもいっしょくたのケースもあるしなあ
いいかげんにしろおおお
OUCH
みとめたくないものだなっ わかさゆえのアヤマチとゆーものを…
モバイル機器ここが困った!?
メモリやハードディスク容量が足りない
ノートPC
ポケットPC
画像データのやりとりをしたい
デジカメ
撮影枚数をふやしたい
PDA
記憶容量が少ない／データを保存したい
スケジュールやメモなどのデータのやりとりをしたい
ケータイは？
うーんコレは何とも…
機種変更かな～
データのやりとりなら赤外線通信とかUSBとかいろいろあるわよねー？それじゃダメかしらん
もちろんそれもアリじゃが，少し面倒なのと，古い機種だとUSBとかサポートがなかったりするしな
IEEE1394とゆーのも流行だよね
するとやっぱりメモリカードとかって便利なのかしら？
取りはずしも自由でディスクとかのメディアがわりにも使えるわよねー？
そう！使えるモノだ!!今回はモバイル機器中心にメモリカードや外部記憶装置をまとめてみた!!

モバイル機器向け外部記憶装置!!

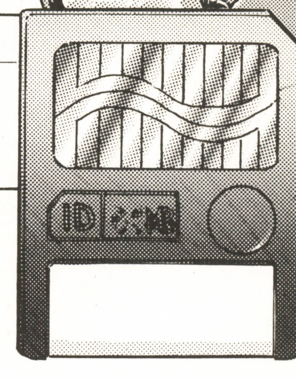


メモリカード型（容量は16～128MBだが今後ふえる）



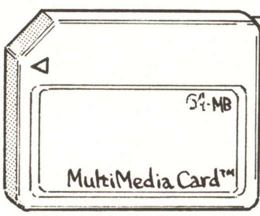
コンパクトフラッシュ（CF, Compact Flash）

元祖？フラッシュメモリ（電池なしでも消えないメモリ）採用のメモリカード。外部インターフェースがPCカードと同じなので，CFアダプタカードによってほとんどのノートPCで使え，多くのPDAやデジカメでも使われている。



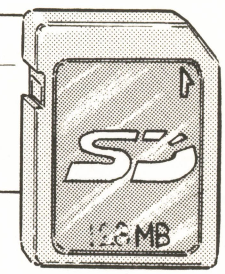
スマートメディア（Smart Media）

一見大きそうだが薄さではNo.1（0.76mm）！CFと双璧をなすPDAやデジカメのメモリカードだ。パソコンとはフロッピーディスク型アダプタのフラッシュパスを使う。著作権保護用にID付タイプもある。



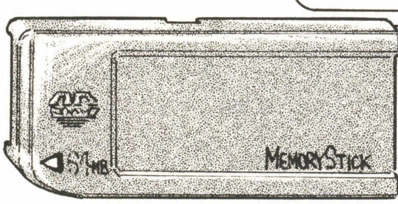
マルチメディアカード（MMC, Multi Media Card）

モバイル機器のうち，とくに携帯電話での利用を考えて開発されたメモリカード。こちらも著作権保護用のSMMC（Secure MMC）がある。パソコン用にPCカードアダプタがあるが，スマートメディア用と同じくやや割高。



SDメモリカード（Secure Digital Memory Card）

MMCの上位互換で，最初から著作権保護を考慮したメモリカード。外部インターフェースもMMCと同じだが，MMCの厚さ1.4mmに対して2.1mmなので，必ずしも物理的に使えるとは限らない。家電メーカー主導で，これからの展開が見ものだ



メモリースティック（Memory Stick）

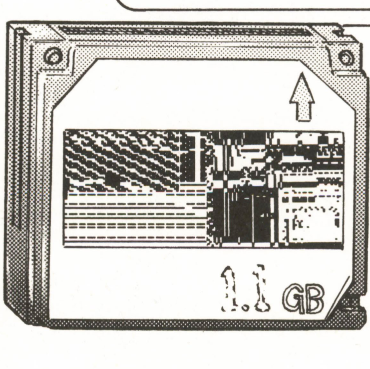
ごぞんじソニーのモバイル機器からPCまで何でも使えるメモリカード。著作権保護用のMagic Gate対応のものや，長さが半分サイズのDuoと呼ぶものもある。



サムドライブ（Thumb Drive）

変わりダネだが，USBポートに差し込むので今どきのPCやPDAほとんど何でも使える！ドライブと名前にあるが中身はフラッシュメモリだ。同種の応用製品に期待しよう！

ハードディスク型



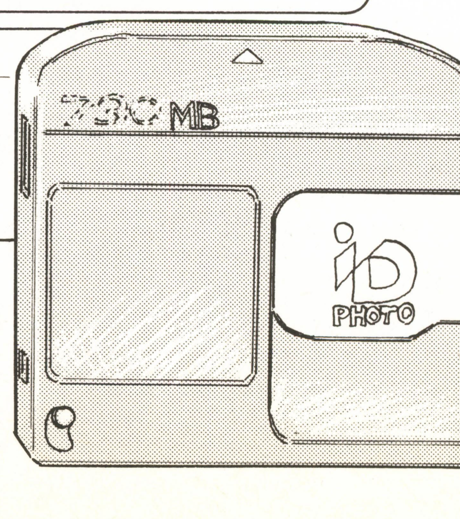
マイクロドライブ（Micro Drive）／コンパクトスター（Compact Star）

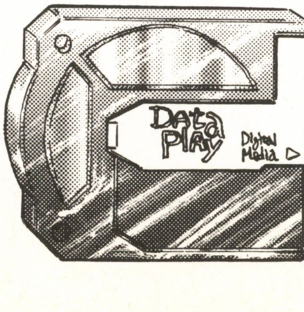
CFとまったく同じサイズ，外部インターフェースの超小型HD。なので，PDAはもちろんPCでもPCカードアダプタ経由で使える。Micro Driveで1GB，Compact Starで1.1GBが最大だが将来的に5～6GBまでふえる技術も開発されている。

光磁気ディスク型

iD PHOTO

小型の光磁気ディスク。デジカメの用途を主に作られた。730MBの容量があるが，高速書き込みが可能で耐久性も高いぞ。





Data Play

まだ市販されていない（2001年秋？）が，この大きさで500MBの容量をもつ。MP3なら11時間分，MPEG4動画なら2時間記録できるが，果たして…!?



メモリカードの種類による容量くらべはあまりイミがなくてな…
どっちかというとどのくらい広まっているか，使いたいモバイル機器で採用されているか，ね
SDもSMもCFもにてるし—
CF
SD
USB
ま，PDAなんかだと何でも使えるのもあるがな
容量に関していえば，やはり最強なのはメモリカードと同じ大きさ，インターフェースの超小型ハードディスクだ!!
現在はまだ1GB強が最大だが，最新のAFCという技術を使えばナンと6GBも夢ではないのだっ!!
ほえ——カードサイズでそれなら，フツーのサイズのHDだったらいったいどれだけ…
うむ！デスクトップPCでTB（テラバイト）の容量すら現実的になったのだ!!
まーとりあえず今は1GBも容量があれば十二分よ!! 某社のPDAなんかはLinuxをOSとして動かせるそうだし…
ポケットPCや高機能PDAはまさしくモバイルプログラミング環境になれるってわけね!!
いつでもどこでもプログラミング♡
ただし，そうは言ってもとくにハードディスクは衝撃には弱い!! 不安は残る…
そこで保存には光磁気ディスク系のメディア，大量バックアップにはハードディスクふだんはメモリカード…と使いわけるとか？
ま，使いわけるなら，整理・分類用にケースをオススメするわよ
耐衝撃も考えた，クールなケースもあるの!!
そうそう，つぐ美ちゃん的にはメモリ増設もたくさんできる通信カートリッジがほしいかしらー？
通信カード
メモリ
メーカーさんつくってくれないかしらー？
む？
そんなアナタにプレゼント!! 影さん特製…
究極のなんでもカートリッジ!!
おおおっ
…と，これ，全然何にもつかないんだけど？
うわあああっ しまったあああっ
To Be Continued...

テクニカルニューススポット
Technical News Spot

プログラム開発環境が各社より続々と新登場！

テクスポ　2001年7・8月合併号

本連載のみ合併号のため（苦しい言い訳）1か月休んでいる間に，新しい開発ツールが続々と登場していました。今回は，ボーランドの『Kylix』『Delphi 6』『JBuilder 5』，そしてマイクロソフトの『Office XP』についての情報をお届けします。

TEXT●石川直人



Delphi使い。原チャリ通勤からステップ・アップするために，普通二輪免許取得をめざして教習所に通う毎日。免許を取る前に，バイク，レインウエア，新しいヘルメットとひとそろえてしまった。後は免許だけなのに。

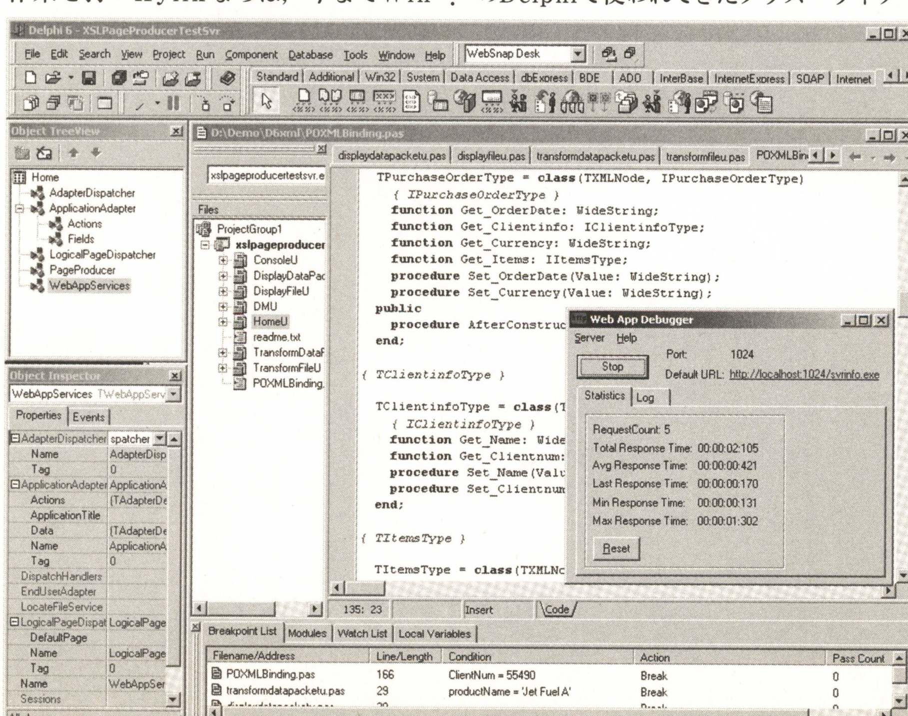
ニュース

Kylix発売&Delphi 6発表！

この連載でも何度か取り上げた，ボーランドのLinux用開発ツール『Kylix日本語版』がついに発売されました。Kylixを使えば，LinuxのX-Window環境でWindowsのDelphi（やMicrosoft Visual Basic）のような快適な開発環境が利用できます。

限りなくDelphiに近い開発環境／言語体系を持つKylixならば，今までWindows用プログラムしか作ったことがない人でも（Linuxのセットアップができれば）簡単にLinux用GUIプログラムが開発できるようになるのです。なお，価格は『Kylix Desktop Developer』が68,000円です（平成13年9月20日までの期間限定価格，Delphiユーザーはさらに優待割引あり）。

そして，Windows用開発ツールである『Delphi』も，最新バージョン6が発表されました（**写真1**）。Delphi 6では，従来のDelphiで使われてきたクラス・ライブラリVCLに加え，Kylixで新たにサポートされたクラス・ライブラリCLXにも対応しています。CLXベースで開発したアプリケーションならば，まったく同じソースコードがDelphiでもKylixでもコンパイルできるようになるのです。

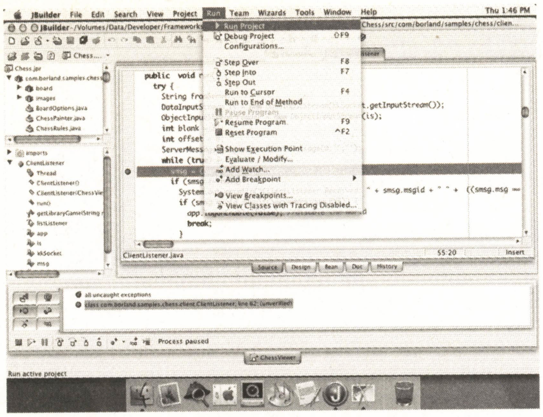


《写真1》Delphi 6を使えばLinux／Windows両用アプリケーションの開発も夢じゃない！　なお，価格はまだ未定だ

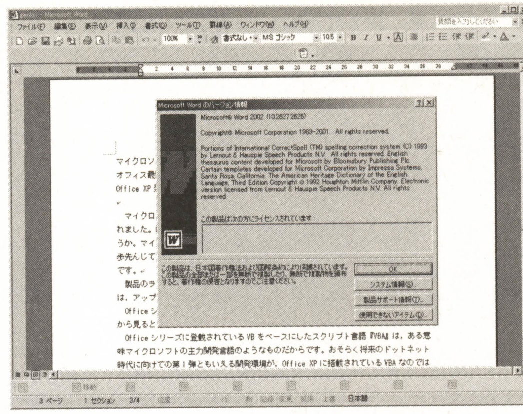
ただし，いくら開発ツールが同じとはいえ，動作環境であるOSがちがう部分に対しては，プログラマーがあらかじめ対応するコードを書いておかなければなりません。Delphi／Kylixを使ってWindows／Linux両対応のアプリケーションを作成しようと考える人は，以下のようなポイントに気をつける必要があります。

- **Windows（Win32）APIはLinuxでは使えない**
- **その他Windows独自のコントロール（OCX）や連係機能（OLE）などもKylixでは使えない**
- **WindowsとLinuxではファイルシステムがちがう（大文字小文字の扱いなど）**

また，Kylixで開発できるのはGUIアプリケーションだけではなく，Webアプリケーションの開発にも対応しています。「Kylix＋Apache（Linux上で動作するWebサーバ）」で作成したWebアプリケーションは，ほんのちょっと書き替えるだけで「Delphi 6＋IIS（Windows上で動作



《写真2》Mac OS X上で動作するJBuilder 5ベータ版。Mac OS Xのインターフェース上で見るJBuilderは，かなり新鮮。価格は未定



《写真3》TVコマーシャルなどでもおなじみの，Office XP。プログラマーならOffice XP Developer版に注目だ！

するWebサーバ)」でも動作するのです。

OSの垣根を越え，CUI，GUI，Webアプリケーションまで開発できるKylixとDelphiは，これからのコンピュータを変えていくだけの力を持つ製品でしょう。

■問い合わせ先

ボーランド ☎03-5350-9380

http://www.borland.co.jp/

ニュース

JBuilder 5発表 今度はMac OS Xにも対応！

ボーランドの簡単アプリケーション開発ツール3兄弟，『Delphi（Kylix）』『C++Builder』『JBuilder』の中で，もっともその動作環境OSが多いJBuilderの最新バージョン5が発表されました。

もともとWindows／Linux／Solarisに対応していたJBuilderですが，最新バージョンではついにMac OS Xにも対応するそうです（**写真2**）。まあ最新のMac OS Xは，従来のMac OSとはちがってUNIXをベースに開発されているため，もともとUNIX系OSに対応していたJBuilderが対応するのはごく自然な成りゆきなのですが。

JBuilder 5では，XMLに関する開発機能が完全に統合され，Javaベースの高度なWebアプリケーション開発への対応がより強化されているそうです。

JBuilderシリーズには，無料で使えるFoundationバージョンが用意されているので（JBuilder 5でどうなるかは未定），興味を持った人はぜひ入手してためしてみましょう。

●http://www.borland.co.jp/jbuilder/foundation/

ニュース

マイクロソフトよりオフィス最新バージョン Office XP発売！

マイクロソフトのオフィス・アプリケーション・スイート最新版『Office XP』が発表されました（**写真3**）。6月8日発売とのことなので，すでに使っている読者も多いのではないでしょうか。マイクロソフトの最新OSである『Windows XP』（英語版10月25日発売予定）に一歩先んじて『XP』ブランド（【eXPerience：経験】から取ったそうだ）がデビューするわけです。

製品のラインナップは，**第1表**のようになっています。一番安価な『Office XP Personal』は，アップグレード版で実売2万円ほどです。

《第1表》Office XPのラインナップ

エディション名	搭載アプリケーション
Office XP Personal	Word，Excel，Outlook
Office XP Standard	Word，Excel，Outlook，PowerPoint
Office XP Professional	Word，Excel，Outlook，PowerPoint，Access
Special Edition	FrontPage，Publisher（Microsoft IntelliMouse Optical付属）
Office XP Developer	Word，Excel，Outlook，PowerPoint，Access，カスタム・ツール

Officeシリーズは，本来はビジネス・アプリケーションなのですが，プログラマーの目から見るとOfficeシリーズは開発ツールであるともいえます。

Officeシリーズに搭載されているVBをベースにしたスクリプト言語『VBA』は，ある意味マイクロソフトの主力開発言語のようなものだからです。おそらく将来のドットネット時代に向けての第1弾ともいえる開発環境が，Office XPに搭載されているVBAなのではないでしょうか。

●問い合わせ先

マイクロソフト ☎03-5454-2300

http://www.microsoft.com/japan/

今月の予想

ボーランドvsマイクロソフトは延長戦に持ち込まれる！

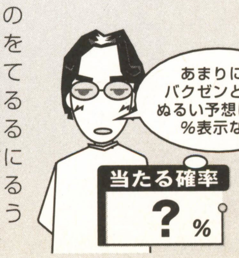
ボーランドがKylixでLinuxでの開発に対応したのに続き，JBuilderではMacintoshにも対応しました。また，もうじきWindowsの次期バージョンであるWindows XPが発売するにもかかわらず，Delphi 6を発表しました。

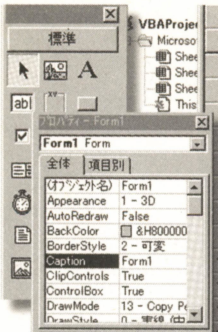
通常Windows用の開発ツールは，新しいOSの発売に合わせて発表／発売されるものです。それを，あえてWindows XPに合わせずに発表してきたあたりに，ボーランドのマイクロソフト離れの傾向が見受けられます。

一方マイクロソフトは，Windows XPより一足早くXPブランドのOfficeを発売してきます。

ドットネットによって，デスクトップOSというものの重要性が薄れていく方向性を示したマイクロソフトにとってみれば，今後重要なのはOSよりもアプリケーション，すなわちOfficeなのかもしれません。

マイクロソフトのドットネット戦略を含めた今後については，その先陣を切る形で発売されるOffice XPの行方によって，見えてくるのではないでしょうか。





はじめよう Microsoft Excelでの BASICプログラミング

協力 富士通南九州システムエンジニアリング

第4回 コントロールを使ってみよう (2)

この連載では，Microsoft Excelを使ってBASICのプログラミングを行います。今月は，先月に引き続きコントロール(部品)の使用方法について解説します。

TEXT & PROGRAMMING 椛島たけし



ワールドカップまで1年を切り，コンフェデ杯の好成績もあり，日本中でサッカー熱が高まっているようだ。ところで，肝心のチケットの抽選結果が送ってこない。かなり不安である。

先月のポイント

まずはコントロールについて大事なところを復習してみましょう。

コントロールとは機能を持つ部品のことで，プロパティ，メソッド，イベントという三つの特徴をもっています。

●プロパティ

コントロールの属性を示す値のことです。属性とは，色，大きさなどコントロールの性質を決めるものです。プロパティは値の設定や取得が可能で，コントロールの性質を知りたいときや変更したいときに使用します。

●メソッド

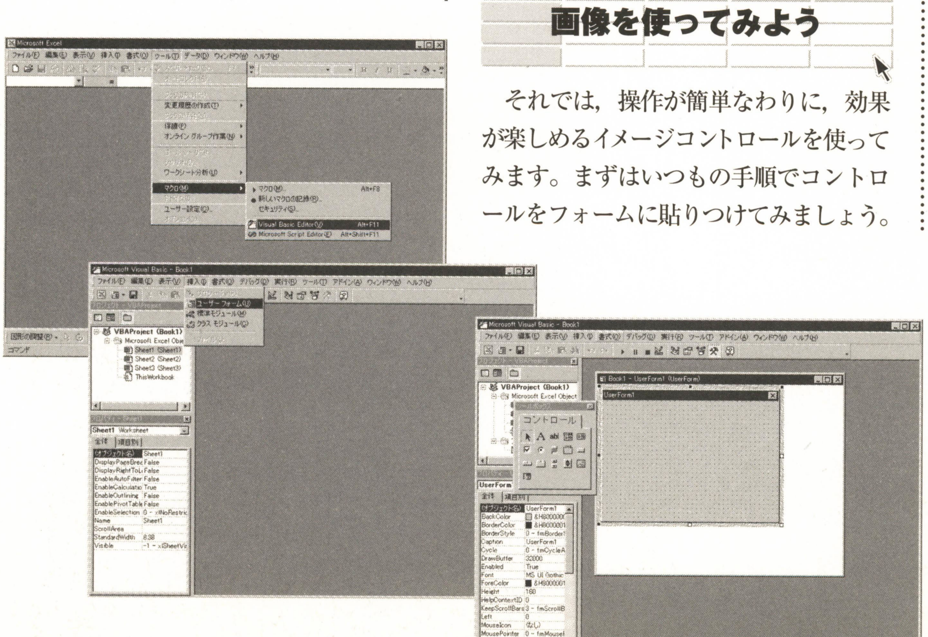
メソッドとはコントロール等がもつ命令のことです。コントロールに備わっている動作を行うときに使用します。つまり，メソッドはコントロールに何かをしてほしいときに使用します。

●イベント

イベントとは，キーを押したり，マウスのクリックなどの操作をしたときに自動的に発生する合図のようなものです。操作だけでなく，プログラム自体がイベントを発生することもあります。イベントは「コントロールが動作したときに行う手続き」に使用します。

画像を使ってみよう

それでは，操作が簡単なわりに，効果が楽しめるイメージコントロールを使ってみます。まずはいつもの手順でコントロールをフォームに貼りつけてみましょう。



《第1図》Visual Basic Editorを起動し，フォームを作成する。もうおなじみの手順だ

Step1

Microsoft Excelを起動し，メニューの [ツール] [マクロ] [Visual Basic Editor] を選択します。

Step2

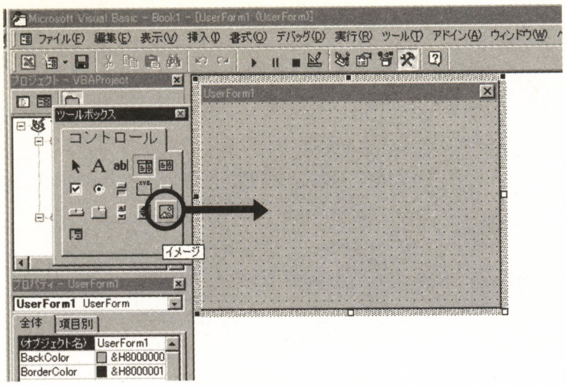
[挿入] メニューから [ユーザーフォーム] を選択します（第1図)。

Step3

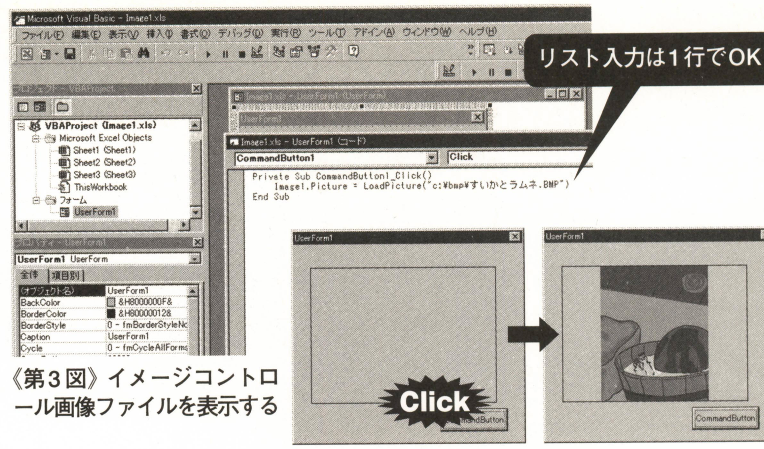
ツールボックスのイメージコントロールをクリック後，UserForm1をクリックし(ドラッグ&ドロップでもよい)，UserForm1にイメージコントロールを貼りつけてください（第2図)。

イメージコントロールに画像を表示するには，Pictureプロパティに画像を設定する必要があります。通常，画像はファイルとして保存されています。LoadPicture関数を使用することで，表示したい画像ファイルを，Pictureプロパティに設定することができます。

画像ファイル（イメージファイル）にはたくさんの形式がありますが，イメージコントロールではビットマップ・ファ



《第２図》イメージコントロールを貼りつけよう



《第３図》イメージコントロール画像ファイルを表示する

イルをはじめ多くの画像ファイル形式がサポートされています（サポートされているフォーマット：＊.bmp，＊.cur，＊.gif，＊.ico，＊.jpg，＊.wmf）。

では，LoadPicture関数を使用してみましょう。

Step4

UserForm1にコマンドボタンを貼りつけ，コマンドボタンをダブルクリック（もしくはコマンドボタンを右クリック後，[コードの表示]を選んでもOK）して，コードウィンドウを表示してください。

ここで，コードウィンドウに表示されるコマンドボタンのクリックイベントに

```
Private Sub CommandButton1_Click ()
      Image1.Picture = LoadPicture
("画像ファイル名")
      'Image1.Picture = LoadPicture
("c:¥bmp¥すいかとラムネ.BMP") ・・
・ 例
End Sub
```

と記述してください。

「画像ファイル名」には表示したい画像ファイルの場所とファイル名を指定します（ドライブ名やフォルダ名を確実に指定します）。適当な画像ファイルがない場合，Windowsをインストールしたフォルダにビットマップ・ファイルが格納されていますので利用するとよいでしょう。プログラムの例が**第3図**となりますので参考にしてください。

プロパティを変更する

イメージコントロールには，表示する画像に関するいくつかのプロパティが用意されています。すこし触ってみることにしましょう。

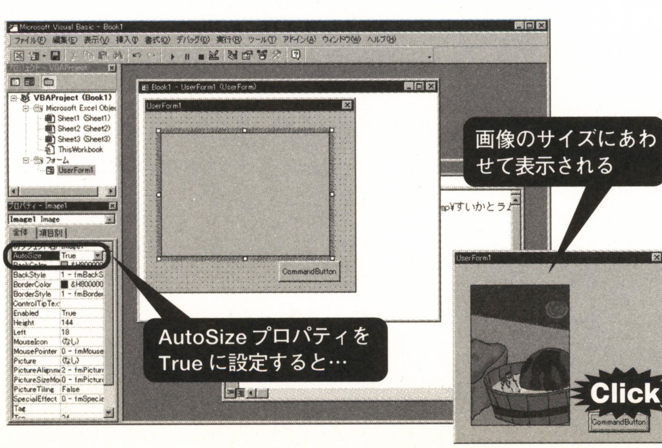
まずは，AutoSizeプロパティです。このプロパティの初期値は"False"ですが，"True"に変更してみましょう。AutoSizeプロパティをTrueに設定すると，画像ファイルの大きさにあわせてイメージコントロールの大きさを調節します。したがって，**第4図**のように，もともとの画像ファイルの大きさで画像が表示されます。

つづいて，PictureSizeModeプロパティを変更してみましょう。まずはAutoSizeプロパティをもとの"False"にもどし，PictureSizeModeプロパティを"1-FmPictureSizeModeStretch"に変更し，実行しましょう（**第5図**）。すると先ほどとは逆に，イメージコントロールのサイズにあわせて画像が調節されます。このように，プロパティを上手に設定することで目的の表示を行うことが可能です。

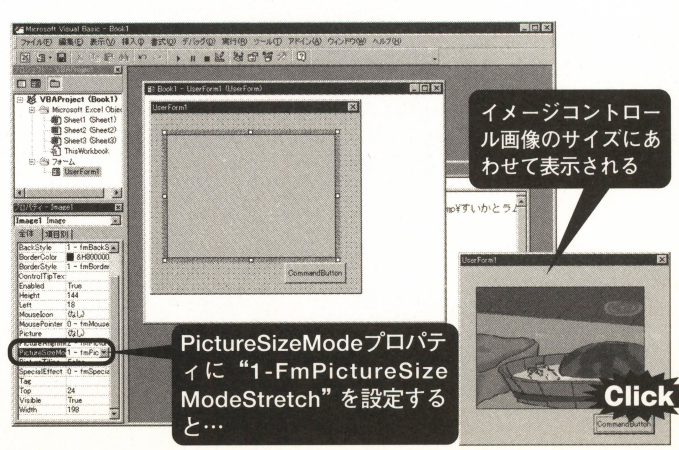
コマンドボタンなどにも使えるぞ

Pictureプロパティに画像を指定すると画像が表示されるわけですが，逆に考えるとPictureプロパティがあるコントロールには画像を表示することができるということです。Pictureプロパティを持つコントロールは，コマンドボタンやラベルなどがあります。

コマンドボタン等へ画像を表示する場合もイメージコントロールと同じくLoadPicture関数を使用します。同じPictureプロパティですので，当然ですよね。簡単にできますのでぜひためしてください（**第6図**）。



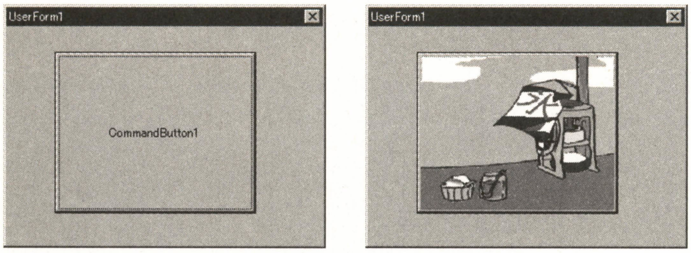
《第４図》イメージコントロールのプロパティを変更してみよう（1）



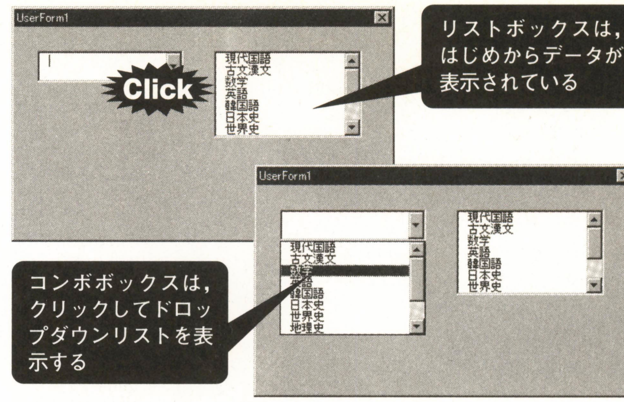
《第５図》イメージコントロールのプロパティを変更してみよう（2）

リスト例

```
Private Sub CommandButton1_Click()
    CommandButton1.Picture = LoadPicture("c:¥bmp¥かき氷.bmp")
End Sub
```



《第6図》コマンドボタンでも画像を表示できるのだ



《第7図》リストボックスとコンボボックスのちがい

リストを選択するには

今度はコンボボックスを紹介しましょう。コンボボックスは，あらかじめ設定された項目から，特定の選択をするときに使用します。リストボックスもよく似た機能を持ちますが，少し機能がちがいます（第7図）。

貼りつけたばかりのコンボボックスには，項目がないため何も選択することができません。コンボボックスに項目を追加するには，AddItemメソッドを使用する必要があります。

コンボボックスとイメージの連携

では，コンボボックスとイメージを連携したプログラムを作成してみることにしましょう。プログラムを起動すると，コンボボックスに画像ファイルのリストが表示され，リストを選択すると選んだ画像が表示されるように工夫してみましょう。まず，画像ファイルを用意します。

Step5

UserForm1にコンボボックスとイメージを貼りつけてます（第8図）。

Step6

UserForm1のコードウィンドウを表示し（方法はStep4を参照），UserFormのInitializeイベントに以下のコードを入力してください。""の中は用意した画像ファイルに関連する名前を指定してください。例では「夏の思い出」に関するファイルを三つほど用意しましたので（第9図），つぎのように指定しています。

```
Private Sub UserForm_Initialize ()
    ComboBox1.AddItem "おいしいすいかとラムネ"
    ComboBox1.AddItem "つめたいかき氷"
    ComboBox1.AddItem "カニもいた夏の海"
End Sub
```

コンボボックスから選択された項目の情報をとりだすには，ChangeイベントでListIndexプロパティを使用します。Changeイベントはコンボボックスで項目が選択された場合に発生します。また，ListIndexプロパティは選択された項目の番号（インデックス）を返します。インデックスは登録順に0からはじまります。したがって，ListIndexプロパティを参照することで，何番目の項目が選択されたかわかるわけです。

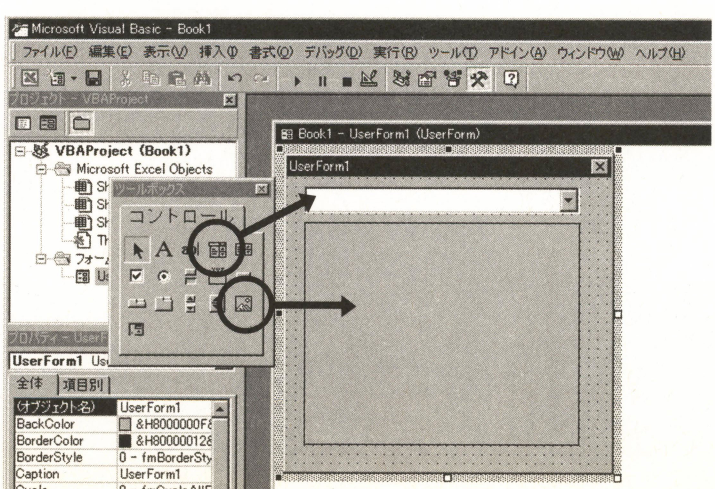
Step7

コンボボックスのコードウィンドウを表示して，Changeイベントに以下のコードを記述してください。

```
Private Sub ComboBox1_Change ()
    Select Case ComboBox1.ListIndex
    Case 0
        Image1.Picture = LoadPicture
("c:¥bmp¥すいかとラムネ.bmp")
    Case 1
        Image1.Picture = LoadPicture
("c:¥bmp¥かき氷.bmp")
    Case 2
        Image1.Picture = LoadPicture
("c:¥bmp¥夏の海.bmp")
    End Select
End Sub
```

Select文は一度学習したので大丈夫ですね？ Step7ではコンボボックスのChangeイベントで，コンボボックスのListIndexプロパティの値（＝選択された項目の番号）をもとに画像ファイルをイメージコントロールにロードしています。入力し終わったら実行しましょう。

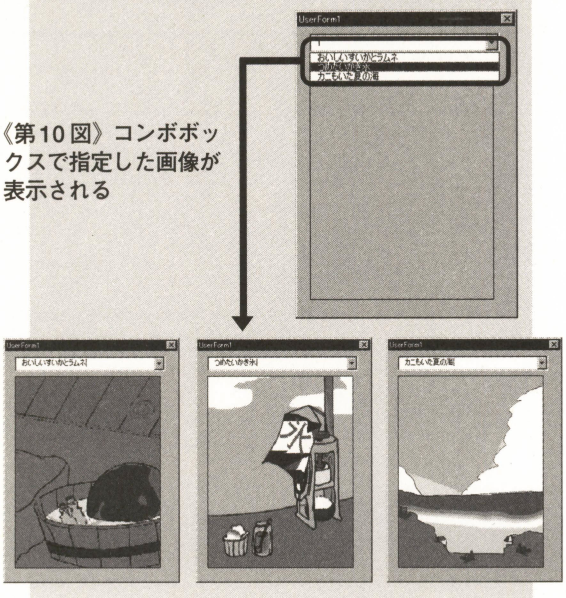
うまくいくとコンボボックスで選択するたびに目的の画像が表示されるはずです（第10図）。



《第8図》コンボボックスとイメージを貼りつけよう



《第9図》夏をテーマにこんな画像を用意してみた



《第10図》コンボボックスで指定した画像が表示される

(プラス面)

- グラフや計算にExcelのシートを利用できる
- シートを利用することでデータの管理や保存が行える

(マイナス面)

- VBのタイマーコントロールのような時間を制御するコントロールがない
- フォームに線や点を表示する機能がない

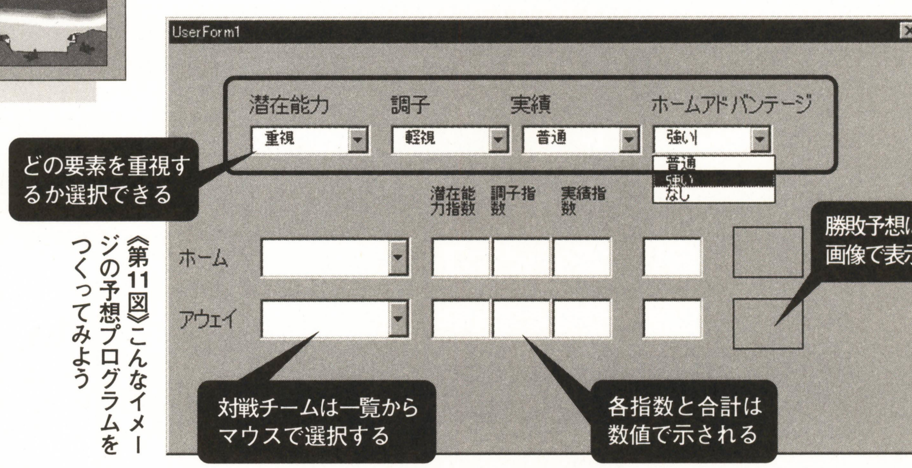
という感じでしょうか。したがって，計算および結果の表示などには向いていますが，動作（アクション）やアニメーション表示を利用するプログラムは苦手といえそうです。

・非常に多くのデータを処理する必要があるシミュレーション
・計算結果をグラフ表示するような処理などが比較的向いているといえます。ゲームでは「五目並べ」，「7つの違い」などアクションを伴わない，思考タイプのゲームが都合良さそうです。

目標，toto予想プログラム

さて，来月からちょっとしたテーマでプログラムを作成したいと思います。Excel VBAの特徴を生かし，かつ趣味性のあるプログラムということで，「toto（サッカーくじ）の予想プログラム」を題材として考えてみました。19歳未満の諸君は買うことが禁止されていますので，せめて予想だけでも盛り上がってみましょう。第11図が完成イメージです。

もちろん，いままで説明した内容を盛り込んでいます。では来月をお楽しみに！



《第11図》こんなイメージの予想プログラムをつくってみよう

この例では，イメージコントロールのPictureSizeModeプロパティに"1-FmPictureSizeModeStretch"を設定しています。

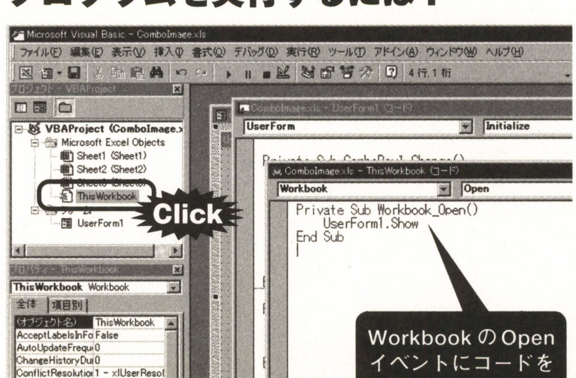
Excelで作成できるプログラムはどんなモノ

さて，2回にわたってコントロールの利用方法について説明しましたが，コントロールの使いかたや機能について理解できたでしょうか？ まだまだ多くのコントロール，プロパティ，メソッド，イベントがありますが，基本をおさえていれば大丈夫です。わからない点はマニュアルを参照することで解決できるはずです。

しかし，Excelにはじめから添付されている（ついている）コントロールは，実際あまり多くはありません。必要最小限といったところでしょうか。これらのコントロールを駆使してプログラムを作成するわけですが，では，どのようなプログラムが向いているのでしょうか？ Microsoft Visual Basicと比較したExcel VBAの特徴を考えると…。

Techinical Column Excel

ブックを開いたときに，プログラムを実行するには？

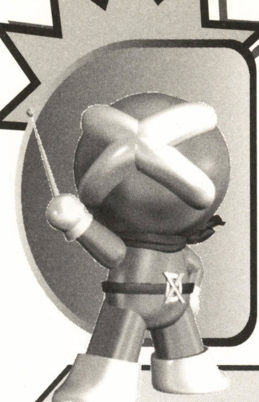


▲Excelブックを起動すると同時にプログラムを起動する方法

Excelブックを開くと同時に，作成したプログラムを実行するにはどうすればいいのか？ プログラムを実行することは多くの場合，フォームを開くことだ。フォームを開くにはUserFormのShowメソッドを使用する。また，Excelブックを開いたときには自動的に，WorkbookのOpenイベントが発生する。したがって，Excelブックを開いたときに，プログラムを実行（フォームを開く）するには，Workbookのオープンイベントで開きたいフォームのShowメソッドを実行すればいいことになる。

マイクロソフトの技術者がDirectXテクノロジーを伝授する！

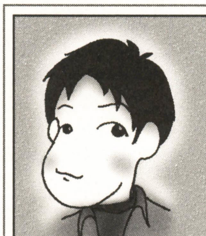
IE+DirectAnimation マルチメディア・プログラミング


▲X君

第5回　サウンド処理で表現力を高めよう！

このコーナーでは，『Microsoft Internet Explorer（以下IEと略）』と『DirectAnimation*1』を使って，誰でも簡単に作れるマルチメディア・プログラムを紹介します。第5回は，サウンドを取り扱うテクニックについて解説し，**写真1**のようなサンプルを作成してみましょう。

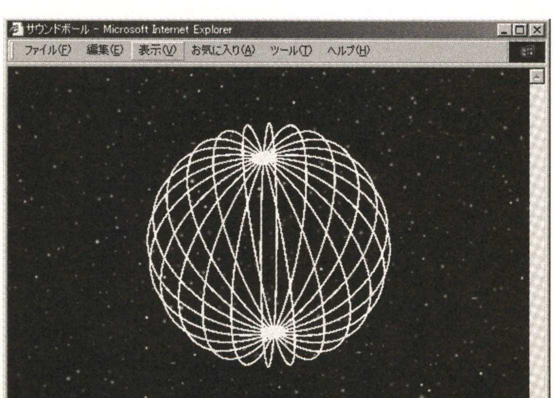
TEXT&PROGRAMING●平井昌人


横浜市生まれの&H25歳男性。1993年4月に，開発言語製品のサポート・エンジニアとしてマイクロソフトに入社。現在はシステム・エンジニア（SE）としてVisual Studioを中心とした開発系の啓蒙・支援活動を行っている。恒例となったDirectXのテクニカル・カンファレンス『メルトダウン』が，7月30日，31日にセンチュリーハイアット東京で開催。私も勉強しに行くかなぁ。

DirectAnimationでサウンドを扱う

DirectAnimationには，グラフィックスだけでなくサウンドを処理する多彩な機能も組み込まれています。DirectAnimationでは，ウェーブ・オーディオ（~.wav），MIDIシーケンス（~.mid）をはじめ，MPEGオーディオ（~.mp2／mp3）など，さまざまなメディア・フォーマットをサポートしています。これまでに紹介してきたイメージのアニメーションに，サウンドを組み合わせて一味ちがった演出を引きだすことが可能になるわけです。

今回は，**写真1**のように，グラフィックスのアニメーションに，バッググランド・ミュージック（BGM）の演奏とイメージの動きに連動させた効果音を付け加えたサンプルを作ってみます。



《写真1》今回作成するサンプル『サウンド・ボール』

DirectAnimationで使えるサウンド・ファイルは？

DirectAnimationでは，ほとんどのサウンド・ファイルがサポートされています。また，サウンドが含まれる動画ファイル（MPEG，AVI，MOV，QTなど）も利用することが可能です。

DirectAnimationでは，**第1表**の形式のサウンド・ファイルをサポートしています。『Microsoft Windows Media Player（以下Media Playerと略）*2』で再生できるメディアであれば，ほとんどの場合OKといえます。ただし，CDオーディオは不可なので，Media PlayerでWindows Mediaオーディオ ファイル形式（~.wma）に変換してください。

サウンド基本テクニック

それでは，「DirectAnimationでサウンド・ファイルを読み込んでそれを再生する」という基本的な手順を紹介します。サウンドを取り扱ううえで，基本的なパターンとなる部分です。

サウンドの管理はDASoundオブジェクト

まずサウンドの制御に関しては，DASoundオブジェクトが担当します。実行時のアニメーションでは，イメージとサウンドが一体化しているように見えます。しかしプログラミング上は，イメージはイメージ（DAImage），サウン

＊1　DirectAnimation
2Dや3Dグラフィックス，サウンド，映像などを総合的にサポートする高レベルなAPIを提供している。IEに統合されており，インターネットでの利用を強く意識した設計となっている。

＊2　Microsoft Windows Media Player
豊富な機能を満載し，操作性を重視したオールインワンのデジタル・メディア・プレーヤ。デジタル・メディアを再生するだけでなく，ジューク・ボックス機能を使った音楽CDのコピー，オーディオ，ビデオのストリーム再生，デジタル・メディア・ファイルの整理，インターネット・ラジオ，最新携帯型プレーヤのサポートなど，数々の機能が搭載されている。また，プレーヤの外観やデザイン（これをスキンという）を根本的に変えてしまう機能もある。SDKを入手すれば，自分でスキンを作成することも可能だ。Media Playerの最新バージョンは，以下から無償で入手できる。
●http://www.microsoft.com/japan/windows/windowsmedia/download/default_1.htm

＊3　絶対パス
ルート・ディレクトリから，省略なしに目的のフォルダやファイルの位置をたどって記述したパス。
●"C:¥Inetpub¥wwwroot¥Sound¥sample.wav"
●"http://www.basicmagazine.net/Sound/foo.mid"

＊4　相対パス
自分のいる位置（カレント・ディレクトリ）を基準として，そこから相対的にたどって記述したパス。自分から見て，上のフォルダや下のフォルダのようにたどっていく。
●"..¥Images¥sample.wav"
●"/Sound/foo.mid"

ドはサウンド（DASound）というように，わけて処理しなければなりません。

DirectAnimationでサウンド機能を利用するためには，つぎのようにDASoundオブジェクト変数を宣言しておきます。sndがDASoundオブジェクト変数です。

```
Dim snd As DASound
```

サウンド・ファイルのロード～ImportSoundメソッド

DirectAnimationでサウンド・ファイルを読み込むためには，ImportSoundメソッドを使います。書式は**第2表**のようになり，引数に読み込むサウンド・ファイル名を指定します。

DirectAnimationにはファイルを読み込むメソッドがいくつか存在しますが，すべて「Import～」という接頭語が付いているのでわかりやすいと思います。ファイル名は，有効なパスまたはURLを付けて指定します。絶対パス*3でも，HTMLファイルからの相対パス*4でもOKです。なお，ファイルをHTMLファイルと同じフォルダに置いておけば，ファイル名だけでOKです。

つぎのコードは，DirectX SDKに含まれている“canyon.mid”というMIDIファイルをロードして，DASoundオブジェクトを生成します。

```
Set snd = m.ImportSound("canyon.
mid").Sound
```

ここで，ImportSoundメソッドの後に，Soundが付いていることに気づくと思います。ImportSoundメソッドは，いったんDAImportationResultオブジェクト*5を返すため，Soundプロパティからサウンド部分だけを抽出してDASoundオブジェクトを取得しているのです。

なお，**リスト1**は“canyon.mid”を再生させるもっとも単純なサンプルです。画面には何も表示されませんが，MIDIファイルを再生します。

リピート再生の設定～Loopメソッド／Repeatメソッド

通常，何も設定しなければ，サウンドの再生は1回で終わります。このため，BGMなどのようにリピート再生が必要な場合は，**第2表**に示すLoopメソッドを使います。

また，Loopメソッドの代わりにRepeatForeverメソッドを使っても，同じように無限リピート再生をさせることができます。つぎのコードは，Loopメソッドで無限リピート再生を指定します。

```
Set snd = snd.Loop
```

また，1回とか2回とかリピート回数を指定したい場合は，Repeatメソッドを使います。つぎのコードは，3回リピートして終わります。

```
Set snd = snd.Repeat(3)
```

サウンドはコントロールのSoundプロパティにセット

最後に，サウンド（DASoundオブジェクト）をコントロールのSoundプロパティにセットし

《第1表》DirectAnimationで扱えるファイル一覧

形式	拡張子
ウェーブ・オーディオ（WAVEファイル）	wav
Musical Instrument Digital Interface（MIDIファイル）	mid
Windows Media Audio（WMAファイル）	wma，wax
Moving Pictures Experts Group（MPEGファイル）	mpeg，mpg，mpe，m1v，mp2，mpv2，mpa
MPEG-2 layer 3（MP3ファイル）	mp3
Audio-Visual Interleaved（AVIファイル）	avi
Windows Mediaファイル	asf，asx
Windows Mediaオーディオ／ビデオ・ファイル	wmv，wmp，wmx

《第2表》DASoundオブジェクトに関する主なメソッド

`Set snd = m.ImportSound(url).Sound`	urlで指定されたサウンド・ファイルの読み込み。戻り値はDASoundオブジェクト。urlは，文字列で有効なパスまたはURLを指定。
`Set snd = snd.Loop`	無限リピート再生の設定。
`Set snd = snd.RepeatForever`	無限リピート再生の設定。Loopメソッドと同じ。
`Set snd = snd.Repeat(n)`	リピート回数の設定。nはLong型でリピート回数。
`Set snd = snd.Gain(vol)`	音量（ボリューム）の設定。volはDouble型の数値で，0.0～1.0の範囲。[0.0]最小←→[1.0]最大となる。
`Set snd = snd.GainAnim(num)`	音量（ボリューム）の設定。numはDANumber型の変動値で，0.0～1.0の範囲。[0.0]最小←→[1.0]最大となる。
`Set snd = snd.Pan(p)`	左右のバランス（パンニング）の設定。pはDouble型の数値で，－1.0～1.0の範囲。[－1.0]左最大←[0.0]左右均等 →[1.0]右最大となる。
`Set snd = snd.PanAnim(num)`	左右のバランス（パンニング）の設定。numはDANumber型の変動値で，－1.0～1.0の範囲。[－1.0]左最大←[0.0]左右均等→[1.0]右最大となる。
`Set snd = snd.Rate(pitch)`	再生速度の設定。pitchはDouble型の数値で，1.0を基準にスピード・アップ／スローダウンする。スロー再生←[1.0]ノーマル再生→早送り再生となる。
`Set snd = snd.RateAnim(num)`	再生速度の設定。numはDANumber型の変動値で，1.0を基準にスピードアップ／スローダウンする。スロー再生←[1.0]ノーマル再生→早送り再生となる。
`Set snd = m.Mix(s1, s2)`	DASoundオブジェクトs1，s2のミキシング（合成）。
`Set snd = m.MixArray(sounds)`	複数のDASoundオブジェクトのミキシング（合成）。soundsは，DASoundオブジェクトの配列または，Array関数によるバリアント配列。

※mはDAStaticsオブジェクト，sndはDASoundオブジェクト。

＊5　DAImportation Resultオブジェクト
読み込んだメディアの情報やデータを，総合的に管理するオブジェクト。サウンドや動画ファイルを読み込むImportSoundメソッドや，ImportMovieメソッドの戻り値がこのオブジェクトである。DAImportation Resultオブジェクトには，メディアの再生時間を取得するDurationプロパティや，DASoundオブジェクトを抽出するSoundプロパティ，動画イメージを取りだすImageプロパティが実装されている。

＊6　音量（ボリューム）
とくにMIDIファイルを再生する場合，DirectAnimationが内部でMIDI（シンセサイザ）のボリューム・レベルを最大まで引き上げてしまう。このため，MIDIファイルを再生したときに，いきなり大音量でビックリさせられるときがある。MIDIファイルを再生する前には，マスタ・ボリュームのレベルを十分に下げておこう。

＊7　サウンドの再生速度
再生速度を調整するRateメソッドは，2.0くらいでかなり速く感じる。MIDIファイルであれば，再生速度を変更しても音程（周波数）には影響をあたえない。しかし，ウェーブ・オーディオやMP3ファイルなどは，再生速度の変化にともなって音程（周波数）も上下してしまうので，注意しよう。

＊8　サウンドのミキシング（合成）
DirectAnimationでは，二つ以上のMIDIファイル（~.mid）を合成して出力することができない。MIDIファイルとウェーブ・ファイルを組み合わせるようにする。もちろん，ウェーブ・ファイルならいくらでも合成できる。BGMとしてMIDIファイルを使い，効果音として複数のウェーブ・ファイルを組み合わせるのが理想的。

特別記念品付き
DirectAnimation
プログラム投稿受付中！

みなさんからのDirectAnimation／DirectX for Visual Basicプログラムの投稿をお待ちしております。投稿されたプログラムは，マイクロソフト平井さんに審査していただき，優秀作品は特選投稿プログラムコーナーに掲載されます。なんと，規定原稿料のほかに，マイクロソフト社から提供される『Visual Studio 97オリジナルTシャツ』をプレゼントします！　どしどしご応募ください。

■投稿要項
DirectAnimationを使ったプログラムは，できるだけVBScriptで書かれたものが対象です（どうしてもできない場合はVBも可）。DirectX for Visual Basicは，VBのソースおよびexeファイルを同送してください。画像，サウンドなどは，著作権および肖像権などを侵害しないように注意しましょう。
投稿のきまりは，特選プログラムコーナーに準じますが，電子メールでも受け付けます。

■あて先
〒141-8715東京都品川区東五反田1-11-15電波新聞社マイコンBASICマガジン「特選プログラムコーナーDirectAnimation」係
電子メール：webmas@basicmagazine.net

ます。

ちょうど，イメージ（DAImageオブジェクト）をコントロールのImageプロパティに代入するのと同じ考えかたです。

```
DAControl.Sound = snd
```

サウンド応用テクニック

サウンド・ファイルをただ再生するだけでは，簡単すぎておもしろくありませんね。

ここでは，音量や左右のバランスの調節，イメージの動きに合わせた変化など，DirectAnimationならではの機能を使ってサウンドを加工するテクニックを紹介しましょう。

音量の変更～Gainメソッド／GainAnimメソッド

音量（ボリューム）[＊6]は，GainメソッドまたはGainAnimメソッドを使います。引数の値は0.0～1.0の範囲で，[0.0]が最小，[1.0]が最大です。つぎのコードは，音量を0.5（50%の出力レベル）に変更します。

```
Set snd = snd.Gain(0.5)
```

また，GainAnimメソッドにDANumber型の変動値をあたえれば，動的に音量をコントロールすることが可能です。サウンドのフェード・イン／フェード・アウトは，これで実現できます。

つぎのコードは，0.0から1.0へ，5秒で変化する変動値（num）を定義してGainAnimメソッドにあたえています。再生すると，じょじょに音が大きくなり，5秒で最大出力に達します。

```
Set num = m.Interpolate(0, 1, 5)
Set snd = snd.GainAnim(num)
```

左右のバランスの設定～ Panメソッド／PanAnimメソッド

左右のバランス（パンニング）は，PanメソッドまたはPanAnimメソッドを使います。引数の値は，－1.0～＋1.0の範囲です。[0.0]で左右均等，[－1.0]で左100%出力，[＋1.0]で右100%出力となります。

つぎのコードは，右に100%出力します。この場合，音は右のスピーカからしか聞こえません。

```
Set snd = snd.Pan(1)
```

これもまた，PanAnimメソッドにDANumber型の変動値をあたえ，パンニングを動的にコントロールできます。右から左へ移動するイメージに合わせて，サウンドの出力も右から左へパンニングさせるような場合に使います。

再生スピードの設定～Rateメソッド／RateAnimメソッド

DirectAnimationでは，再生時のスピードも制御することができます。DASoundオブジェクトのRateメソッド，またはRateAnimメソッドを使います。ただし，逆の再生（巻き戻し）はできません。つぎのコードは，サウンドの再生速度[＊7]を1.5倍に変更して早送りのように再生します。

```
Set snd = snd.Rate(1.5)
```

サウンドのミキシング（合成）

DirectAnimationは，イメージを合成するのと同じく，サウンドも合成することができます。バック・グランド（BGM）でMIDIファイルを再生しておき，あるタイミングで効果音としてウェーブ・ファイルを鳴らすような場合，BGMと効果音をミキシングして出力させるのです。

サウンドのミキシング～Mixメソッド／MixArrayメソッド

イメージ（DAImageオブジェクト）の合成には，OverlayメソッドやOverlayArrayメソッドを利用しますが，サウンドのミキシング（合成）[＊8]をする場合は，MixメソッドまたはMixArrayメソッドを使います。

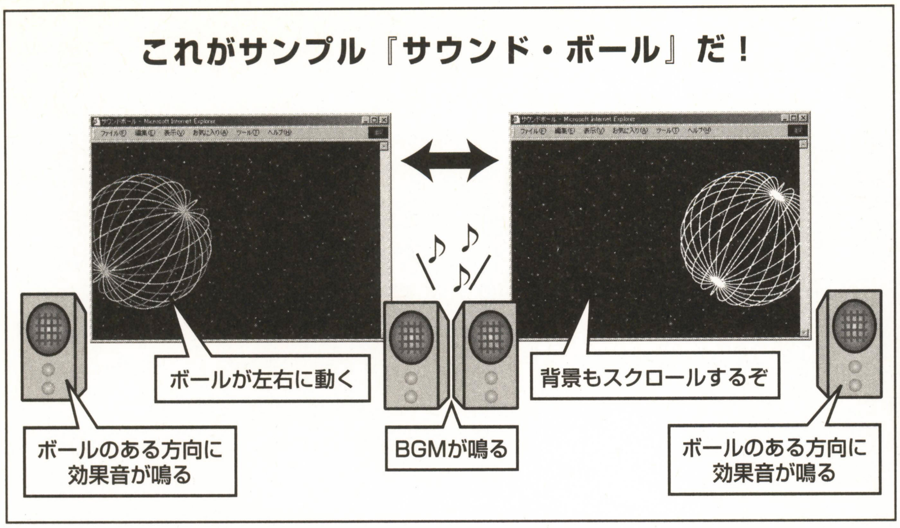
つぎのコードは，Mixメソッドを使ってDASoundオブジェクトのeffとbgmをミキシングして合成したDASoundオブジェクト（snd）を生成します。イメージの合成とちがい，引数の順番は関係ありません。

```
Set snd = m.Mix(eff, bgm)
```

なお，二つ以上のDASoundオブジェクトを合成する場合は，つぎのようにMixArrayメソッド使います。

```
Set  snd  =  m.MixArray(Array(s1,
s2, s3, bgm))
```

リスト2はサウンドのミキシングをする簡単なサンプルです。別々にパンニングを設定したcanyon.midとlose.wavを合成して出力します。左のスピーカーからcanyon.midのメロディが，右のスピーカーからはlose.wavの音が聞こえま



《第1図》サンプル『サウンド・ボール』は，グラフィックスの動きとサウンドの動きが連動している



《写真2》サンプルの背景に使用している，宇宙空間のJPEGファイル“starts.jpg”

す。

サンプル・プログラム『サウンド・ボール』を実行してみよう！

今回のテーマであるサウンド処理を取り入れたサンプル・プログラム『サウンド・ボール』“SndBall.htm”を用意しました。

ソースコードは，**リスト3**です。なお，サンプルのソースコードや画像ファイル，サウンド・ファイルなどは，以下のベーマガ・ホームページにありますのでダウンロードしてください。

●http://www.basicmagazine.net/

リスト3を実行すると，**第1図**のように左右にスクロールしている背景の上を，複数のリングを組み合わせて作成したボールが3D回転しながら左右に動きます。サウンドは，BGMにMIDIファイルをリピート再生しており，ウェーブ・ファイルを効果音としてボールの動きに合わせてパンニングさせています。

BGMは左右均一に出力しますが，効果音はボールの位置に合わせて左←→右に振れます。ステレオ・スピーカ，またはステレオ・ヘッドホンで聴いてみてください。

サンプルで使用しているサウンド・ファイルは，つぎの二つです。

●GM042.MID[*9]

BGM用の，軽快なメロディのMIDIファイルです。

●lose.wav

ボールの位置に合わせてパンニングさせる，2秒の短いウェーブ・ファイルです（DirectX SDK付属）。

サンプルで使用している画像ファイルは，つぎの一つのみです。

●stars.jpg

写真2のような，宇宙空間をイメージしたJPEGファイルです（DirectX SDK付属）。

また，このサンプルではつぎの二つの定数を使っています。

●**NumOfRings：ボールを形成するリングの数**

●**DRAW_WIDTH：リングを描画するときの線の太さ**

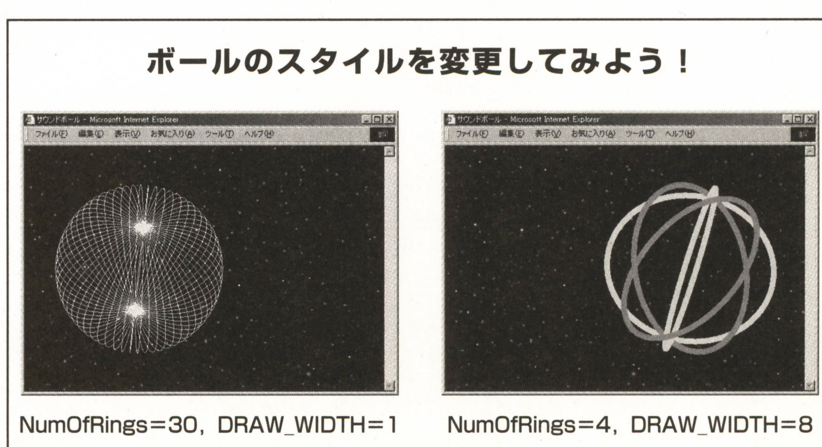
これらの定数は，変更が可能です。たとえば，「NumOfRings = 30」「DRAW_WIDTH = 1」に変更すると，**第2図**の左側のようになります。また，「NumOfRings = 4」「DRAW_WIDTH = 8」に変更すると，**第2図**右側のようになるなど，ボールのスタイルを変更して楽しめます。

リングを組み合わせてボールを作る

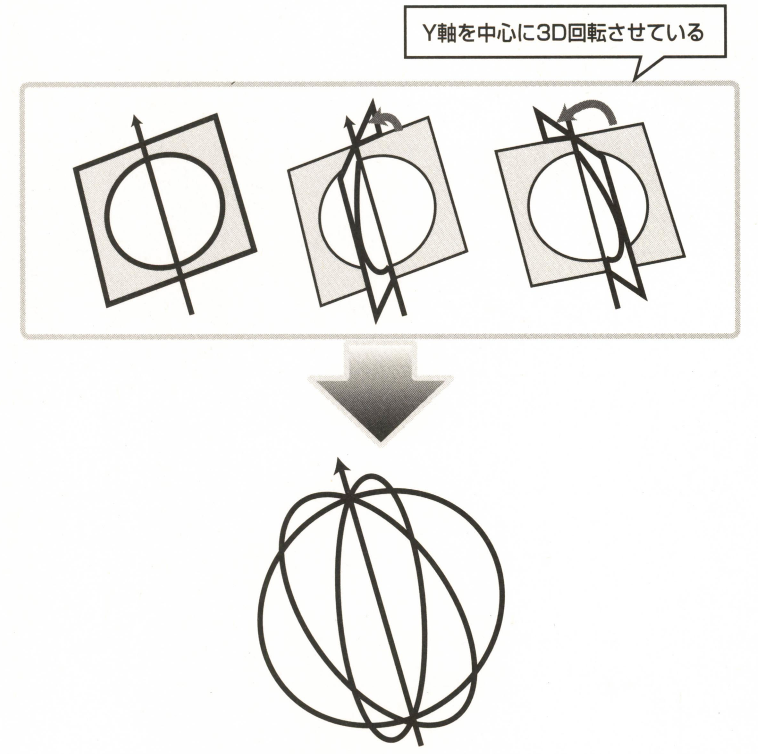
このサンプルのボールのイメージは，**第3図**のように，Ovalメソッドで定義した2Dの円の軌跡（DAPath2オブジェクト）[*10]に3Dの変換を組

＊9　GM042.MID
このサンプルのBGMに使用している“GM042.MID”は，有限会社キュービックエナジー発売の『MusicEnergy5』を使用させていただいた（1998 Cubic Energy Inc. All rights reserved.）。なお，このファイルは2次配布することはできないので注意。

＊10　軌跡（DAPath2オブジェクト）
図形や線，文字などの“軌跡”を管理するオブジェクト。軌跡とは，数学でいう一次変換で表現できる点や線の集合体。点や線を表す方程式のようなものを管理している。簡単にいえば，DAPath2オブジェクトは図形や文字を描画するための“型紙”だ。軌跡（DAPath2オブジェクト）からイメージ（DAImageオブジェクト）を描画するしくみは，型紙に沿って鉛筆でなぞるとその図形が書けるのと同じ。



《第2図》定数NumOfRingsとDRAW_WIDTHの値を変更すると，ボールを描画するライン数が，左側のように増えたり，右側のように減ったりする



《第3図》ボールは，このように円（リング）を組み合わせできているのだ

3D変換を2D変換にコンバートする！

DirectAnimationでは，軌跡（DAPath2オブジェクト）やイメージ（DAImageオブジェクト）に適用できる変換は，これまでの連載で紹介したように2D変換（DATransform2オブジェクト）だけです。

このため，3D変換を2D変換にコンバートする必要があります。それを行ってくれるのが，ParallelTransform2メソッドです。

このサンプルでは，3D回転を定義するRotate3DegreesメソッドとRotate3RateDegreesメソッドを利用しています。構文は，**第3表**のようになります。単位は，“ラジアン*11”と“度”がありますので，注意してください。

3D変換は，DATransform3オブジェクトが管理します。2D変換とちがい，Z軸というものと回転軸を指示するためのベクトル（DAVector3オブジェクト*12）という概念が入ってきます。

3Dの扱いに関しては，近い将来このコーナーでくわしく解説する予定ですので，お待ちください。

み合わせています。つまり，いくつかの円を3D空間上でY軸を中心に均等に3D回転をあたえてから，Drawメソッドでイメージ（DAImageオブジェクト）を作ります。

イメージは，1個1個のリングごとに配列変数（rings）に格納して，OverlayArrayメソッドで一気に一つのイメージに合成しています。

まとめ

さて，今回はDirectAnimationでサウンドの制御に挑戦してみました。単に再生するだけでなく，音量や左右のパンニングまで，思いのままにコントロールできます。

また，紹介したサンプルのように，アニメーションの動きに合わせて変化をつけたり，いろ

＊11　ラジアン
角度の単位で，360度＝2π ラジアン≒2×3.14ラジアンである。
●度（d）からラジアン（r）を求める式
r = d × π ÷ 180
●ラジアン（r）から度（d）を求める式
d = 180 × r ÷ π

＊12　DAVector3オブジェクト
空間（3次元）座標でのベクトルを管理するオブジェクト。DAStatics オブジェクトのVector3メソッドやVector3Animメソッドで作成する。原点（0，0，0）から点（1，2，3）へ向かう3次元ベクトルは，m.Vector3（1，2，3）のように記述する。ちなみに，Vector3（1，0，0）はX軸を，Vector3（0，1，0）はY軸を，Vector3（0，0，1）はZ軸を意味する。これらはm.XVector3，m.YVector3，m.ZVector3というプロパティとして，あらかじめ提供されている。なお，平面（2次元）座標でのベクトルは，DAVector2オブジェクトが担当する。

《第3表》3D回転を定義する主なメソッド

`Set trn3 = m.Rotate3(axis, angle)`
3Dベクトルaxisを回転軸として，angleだけ回転させる。axisはDAVector3オブジェクト，angleはDouble型で単位は“ラジアン”。
`Set trn3 = m.Rotate3Degrees(axis, angle)`
3Dベクトルaxisを回転軸として，angleだけ回転させる。axisはDAVector3オブジェクト，angleはDouble型で単位は“度”。
`Set trn3 = m.Rotate3Anim(axis, num)`
3Dベクトルaxisを回転軸として，変動値numにしたがって回転させる。axisはDAVector3オブジェクト，numはDANumber型で単位は“ラジアン”。
`Set trn3 = m.Rotate3Rate(axis, angle)`
3Dベクトルaxisを回転軸として，毎秒angleで連続回転させる。axisはDAVector3オブジェクト，angleはDouble型で単位は“ラジアン”。
`Set trn3 = m.Rotate3RateDegrees(axis, angle)`
3Dベクトルaxisを回転軸として，毎秒angleで連続回転させる。axisはDAVector3オブジェクト，angleはDouble型で単位は“度”。

※mはDAStaticsオブジェクト，trn3はDATransform3オブジェクト。

いろなサウンドを合成したりと簡単にできるのは，DirectAnimationならではの特長だといえます。

DirectAnimationのように「誰でも・簡単・高機能」と三拍子そろっているお気軽コンポーネントは，他にはありません。まだ一度も動かしてみたことのない人も，だまされたと思ってためしてみてください。

《リスト1》DirectAnimationでサウンドを再生させる一番短いコード“sample1.htm”（VBScript用）

```
<HTML>
<TITLE>MIDIファイルの再生</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">

Option Explicit

Sub Window_onLoad()
  Dim m    'As DAStatics
  Dim snd  'As DASound
  Set m = DAControl.PixelLibrary
  Set snd = m.ImportSound("canyon.mid").Sound
  DAControl.Sound = snd
  DAControl.Start
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト2》サウンドのミキシングをする簡単なサンプル“sample2.htm”（VBScript用）

```
<HTML>
<TITLE>サウンドのミキシング</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">

Option Explicit

Sub Window_onLoad()
  Dim m    'As DAStatics
  Dim s1   'As DASound
  Dim s2   'As DASound
  Dim snd  'As DASound
  Set m = DAControl.PixelLibrary
  Set s1 = m.ImportSound("canyon.mid").Sound
  Set s1 = s1.Loop.Pan(-1)
  Set s2 = m.ImportSound("lose.wav").Sound
  Set s2 = s2.Loop.Pan(1)
  Set snd = m.Mix(s1, s2)
  DAControl.Sound = snd
  DAControl.Start
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

リスト1＆リスト2 入力＆実行方法

このリストは，IE5／5.5とテキスト・エディタ（『メモ帳』など）があれば，実行することが可能だ。IE5は，以下のアドレスから無償でダウンロードすることができる。

● http://www.microsoft.com/windows/ie_intl/ja/download/

リストをメモ帳などで入力したら，拡張子を“htm”または“html”にして保存する。そして，IE5の[ファイル]メニューより，[開く]で保存したファイルを読み込もう。すると，リストが実行されるようすを，IE5で確認することができる。

《リスト3》サンプル『サウンド・ボール』“SndBall.htm”（VBScript用）

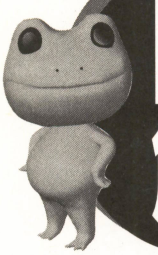
```
<HTML>
<TITLE>サウンドボール</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//-----------------------------------------------
'//
'// DirectAnimation Sample                by Masato
Hirai
'//
'//    『サウンドボール』
'//
'//-----------------------------------------------
Option Explicit

'//-----------------------------------------------
'// 初期化処理
'//-----------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Const NumOfRings = 12            'リングの本数(2以上)
  Const DRAW_WIDTH = 2             '線の太さ
  Dim m          'As DAStatics     'DAStaticsオブジェクト
  ReDim rings(NumOfRings - 1) 'As DAImage 'リングイメージ
の配列
  Dim ring       'As DAImage       'ボールイメージ
  Dim back       'As DAImage       '背景イメージ
  Dim img        'As DAImage       '最終イメージ
  Dim rot3       'As DATransform3  '3D変換(回転用)
  Dim trn3       'As DATransform3  '3D変換(合成用)
  Dim trn2       'As DATransform2  '2D変換
  Dim num        'As DANumber      '変動値
  Dim zero       'As DANumber      'DANumber(0)用
  Dim eff        'As DASound       '効果音
  Dim bgm        'As DASound       'BGM
  Dim snd        'As DASound       '合成サウンド
  Dim pth        'As DAPath2       'リングの軌跡
  Dim border     'As DALineStyle   '枠線
  Dim c1         'As DAColor       'リングの色(パステル系)
  Dim c2         'As DAColor       'リングの色(ホワイト系)
  Dim n          'As Long
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// 変則的な3D回転の定義
  '//
  Set rot3 = m.Compose3( _
    m.Rotate3RateDegrees(m.Vector3(-3, 2, 1), 90), _
    m.Rotate3RateDegrees(m.YVector3, -45))
  '//
  '// リングの色(パステル系とホワイト系)を定義
  '//
  Set num = m.Mul(m.LocalTime, m.DANumber(0.05))
  Set c1 = m.ColorHslAnim(num, m.DANumber(1),
m.DANumber(0.7))
  Set c2 = m.ColorHslAnim(num, m.DANumber(1),
m.DANumber(0.9))
  Set border = m.DefaultLineStyle.Width(DRAW_WIDTH)
  '//
  '// リングイメージの作成
  '//
  For n = 0 To NumOfRings - 1
    '//
    '// 均等にY軸に対して3D回転
    '//
    Set trn3 = m.Rotate3Degrees(m.YVector3, n * 180 /
NumOfRings)
    '//
    '// 3D変換を合成して2D変換へコンバート
    '//
    Set trn2 = m.Compose3(rot3, trn3).ParallelTransform2
    '//
    '// リング(円)の軌跡に合成2D変換を適用
    '//
    Set pth = m.Oval(256, 256).Transform(trn2)
    '//
    '// パステル系とホワイト系の色で交互にリングを描画
    '//
    If n Mod 2 = 0 Then                '← 偶数と奇数で分岐
      Set rings(n) = pth.Draw(border.Color(c1))
    Else
      Set rings(n) = pth.Draw(border.Color(c2))
    End If
  Next
  '//
  '// リングイメージを合成してボールにする
  '//
  Set zero = m.DANumber(0)
  Set ring = m.OverlayArray(rings)  '← リングイメージの
合成
  Set num = m.Sin(m.Mul(m.LocalTime, m.DANumber(0.5)))
  Set trn2 = m.Translate2Anim(m.Mul(num,
m.DANumber(300)), zero)
  Set ring = ring.Transform(trn2)   '← 左右スクロールの
適用
  '//
  '// 背景イメージ
  '//
  Set back = m.ImportImage("stars.jpg")
  Set back = back.Tile                 '← タイリング処理
  Set trn2 = m.Translate2Anim(m.Mul(num, m.DANumber(-
600)), zero)
  Set back = back.Transform(trn2)   '← 左右スクロールの
適用
  '//
  '// イメージの合成(最終イメージ)
  '//
  Set img = m.Overlay(ring, back)
  '//
  '// 効果音の設定
  '//
  Set eff = m.ImportSound("lose.wav").Sound
  Set eff = eff.Loop               '← 無限リピートの設定
  Set eff = eff.PanAnim(num)       '← パンニングの設定
  '//
  '// BGMの設定
  '//
  Set bgm = m.ImportSound("GM042.MID").Sound
  Set bgm = bgm.Loop               '← 無限リピートの設定
  Set bgm = bgm.Gain(0.3)          '← 音量の調節(小さめに)
  '//
  '// サウンドのミキシング(最終サウンド)
  '//
  Set snd = m.Mix(eff, bgm)
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img                '← イメージ
  DAControl.Sound = snd                '← サウンド
  DAControl.Start
  Call Window_onResize()
End Sub

'-----------------------------------------------
'ウィンドウのサイズ変更時の処理
'-----------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

プログラマーをめざす人の必見講座



図解！ Visual Basic ゲーム作成術

第5回　シューティング・ゲームを作る②　複数種類の敵キャラクターを登場させる！

シューティング・ゲームでは，プレイヤーの標的となる“敵キャラクター”が重要な要素になっています。この敵キャラクターの特徴が，ゲームの特徴を大きく左右し，ゲーム自体のおもしろさにも影響をあたえます。このため，敵キャラクターが1種類しか登場しないというゲームは，変化や目新しさに欠け，あまりおもしろいゲームにはなりません。今回は，複数種類の敵キャラクターをゲームに登場させるテクニックを，『Microsoft Visual Basic（以下VBと略）』で実現してみましょう。

TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。最近，インターネット常時接続環境のコストが下がってきた。こまめに切断しながらのアクセスはもうやめようと思い，『フレッツ』にしてみた。コスト的に有利になっていることよりも，落ち着いてアクセスできるところが気に入っている。

Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）

オープン価格

※編集部の価格調査によると，下記価格が目安になります。

ラーニング版2万円強
（アカデミック・パッケージ1万円弱）
プロフェッショナル版8～9万円
（アカデミック・パッケージ1万円強）



問い合わせ先
マイクロソフト
☎03-5454-2300

ホームページ
http://www.microsoft.com/japan/

動作環境
Windows 95／98／2000／NT4.0が動作するマシン

敵キャラクターにどういう特徴を持たせる？

シューティング・ゲームに登場させる敵キャラクターは，そのタイプや登場する数，動きのパターンなどで，さまざまな特徴づけができます。

第1図のように，同時に多数登場し，単純な動きをしてすぐにプレイヤーの弾で破壊されていくようにすれば，そのキャラクターには“雑魚キャラクター”という特徴づけができますね。これに対して**第2図**のように，まれに単体で登場し，複雑な動きでプレイヤー・キャラクターを困惑させるようにすれば，“中ボス・キャラクター”になります。

さらに，ステージの最後に耐久力の高いキャラクターを登場させれば“ボス・キャラクター”，プレイヤー・キャラクターが体当たりするとプレイヤー・キャラクターのダメージが減るようにすれば“回復キャラクター”というように，特徴づけには制限がありません。

「1種類の単純な敵キャラクターが登場するシューティング・ゲーム」と，「多種多様な敵キャラクターが登場するシューティング・ゲーム」を対比してみましょう。ゲームに登場する敵キャラクターにいろいろな特徴を持たせるということが，ゲームのおもしろさを高める有効な方法である，ということがわかると思います。

敵キャラクターの要素をきめよう！

敵キャラクターには，いろいろな要素があります。つぎに示す要素をどのようにきめていくかによって，敵キャラクターの特徴が変わってきます。

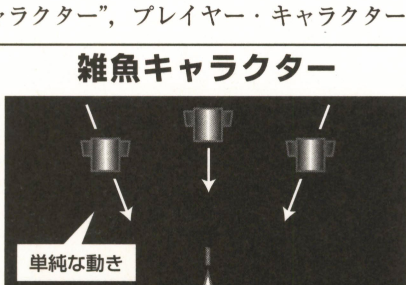
●攻撃力

弾をたくさん発射したり，プレイヤー・キャラクターを正確に狙う弾を発射すれば，攻撃力は高まりますし，弾を出さないようにすれば低くなります。攻撃力を高めすぎると，プレイヤー・キャラクターが必ず撃破されてしまったりして，ゲーム・デザインを崩す場合があるので，注意が必要です。

●耐久力

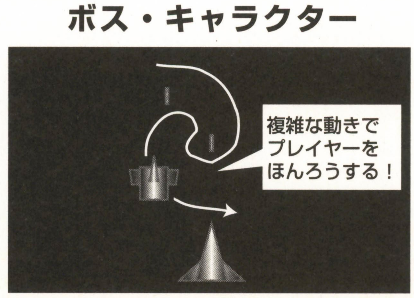
プレイヤー・キャラクターの攻撃にある程度耐えるように設定することで，“堅いキャラクター”という印象をあたえられます。ボス・キャラクターなどは，耐久力を上げておくのが一般的です。ただし，大きな耐久力をあたえる場合は，攻撃パターンなどを変化させ，プレイヤーから

雑魚キャラクター

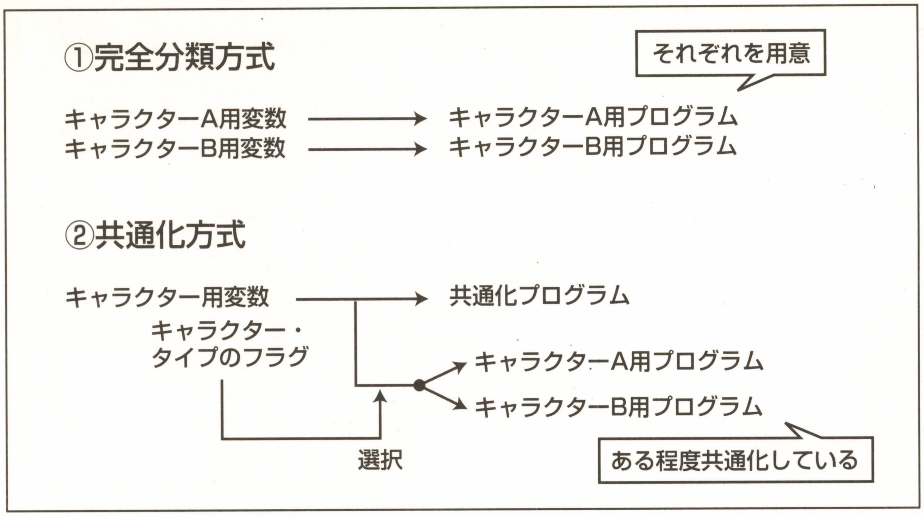


《第1図》単純な動きで複数登場させると，雑魚キャラクターという認識が深まる

ボス・キャラクター



《第2図》ステージの最後など，節目で登場する強いキャラクター。動きのほか，キャラクターの色や大きさなどでもボスらしさを表現できるぞ



《第3図》キャラクターの性格によって，完全分類するか共通化するかわけて考えよう

見て無意味に堅い，という印象を持たれないようにしなければなりません。

●移動速度

移動速度が高いキャラクターは，プレイヤー・キャラクターからの攻撃が当たりにくくなるため，耐久力が低くてもやっかいな敵に見えます。また，画面に急に登場させることで，プレイヤーを驚かせることもできます。ただし，急に登場することでプレイヤー・キャラクターにダメージをあたえると，プレイヤーに対するストレスになります。プレイヤー・キャラクターから遠い場所に急に登場させ，その後はほどほどの速度で移動させる程度にするのが有効です。

●移動アルゴリズム

単純な移動をさせれば，プレイヤーから攻撃しやすくなり，“雑魚”としての特徴を持たせられます。プレイヤーの攻撃を巧みに回避させれば腕の立つ敵キャラクター，という位置づけになります。ただし，これも程度問題で，あまりにも回避性能が高い敵キャラクターは，プレイヤーに対するストレスになります。

●同時登場数

画面中に多数の同一キャラクターが登場すると，そのキャラクターは雑魚キャラ的に見えます。一つだけ登場させれば，ボス・キャラクター的な印象をあたえます。

●登場頻度

何度も登場するキャラクターは雑魚キャラクター的になりますが，同時にそのゲームの定番キャラクターとしても見えます。特徴あるキャラクターをまれにしか登場させないようにすれば，プレイヤーの記憶に残るようになります。

●登場タイミング

ステージの最後に登場させれば，やはりボス・キャラクターになりますね。また，ステージの区切りになるところだけに登場させるようにすれば，ステージの区切りを知らせるキャラクターという印象をあたえることもできます。

ここまで説明したように，敵キャラクターの要素をどのようにきめるかで，そのキャラクターの特徴は大きく変化します。ゲーム・デザインに応じて，キャラクターの特徴を割りあてていく必要があるわけです。

複数種類の敵キャラクターの処理

敵キャラクターの要素を実際に表現するのは，敵キャラクター処理プログラムです。よって，敵キャラクターの要素がきまったら，それを実現するようなプログラムを用意しなければなりません。

“敵キャラクター”という表現では，特徴を持ったいろいろな敵キャラクターすべてが同じようなものになってしまいますが，実際には特徴づけによってかなり異なるキャラクターになります。これらのキャラクターの処理を行う方法は，主につぎの2種類が考えられます（**第3図**）。

①完全分類方式

一つは，キャラクターの種類に応じて，処理プログラムとデータ記録変数を個別に用意する方法です。キャラクターAとキャラクターBが存在する場合を考えてみましょう。この場合，

●キャラクターA用の変数
●キャラクターB用の変数
●キャラクターA用の処理プログラム
●キャラクターB用の処理プログラム

の四つをすべて用意するというものです。

②共通化方式

もう一つは，処理プログラムとデータ記録変数を共通化して，プログラム中でキャラクターの種類に応じた場合わけを行うというものです。

どちらの方式がふさわしいかは，個々のキャラクターの特徴がどの程度似ているか／ちがっているかで判断します。

キャラクターの特徴が似かよっている場合，一つ目の完全分離方式では似たようなプログラムが二つできてしまいます。また，変数も2グループ用意する必要があるため，プログラミングが量的にむずかしくなります。

逆に，キャラクターの特徴が大きく異なっている場合，二つ目の共通化方式では場合わけがやっかいになり，プログラミングが質的にむずかしくなります。

よって，キャラクターの特徴を見きわ

ベーマガ・ホームページ ダウンロードの方法

まずは，以下のアドレスにインターネット・アクセスします。

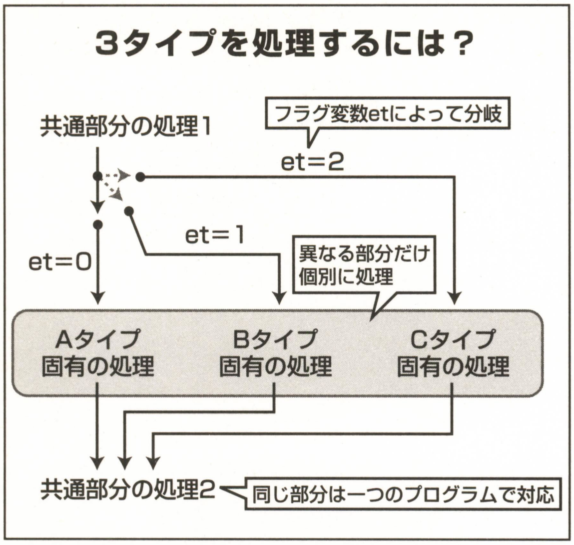
●http://www.basicmagazine.net/

そこから，「連載プログラミング講座」をクリックしてください。

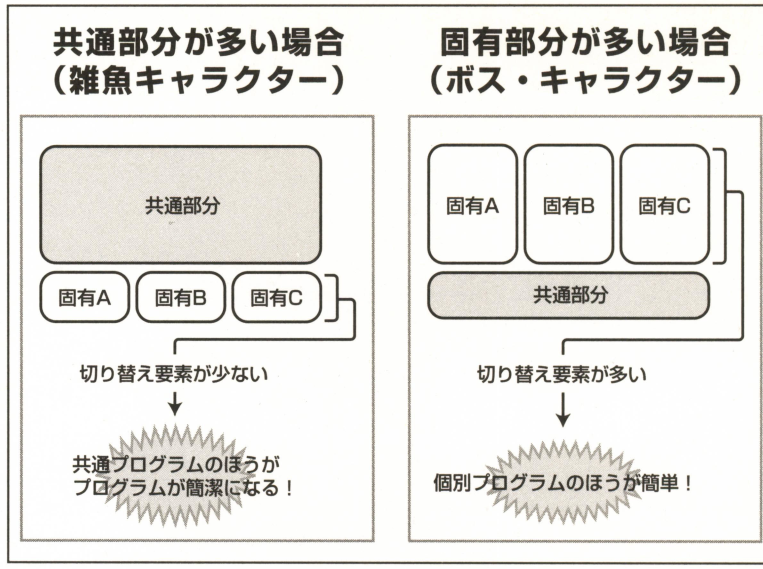
そして，『図解！ VBゲーム作成術』の画面の下にある「ファイルをダウンロード」をクリックします。すると，「ファイルのダウンロード」ウィンドウが表示されるので，「このファイルをディスクに保存する」をチェックして[OK]ボタンを押します。「名前を付けて保存」ウィンドウが表示されるので，適当な場所（c:¥bm08など）に保存します。

同じ場所に“lha.exe”をコピーして，[スタート]-[プログラム]-[MS-DOSプロンプト]で，その場所に移動します（上記の例だと「cd c:¥bm08」と入力）。そして，「lha e z2001-08」とキーボードから入力して，[Enter]キーを押します。これで，解凍作業は終わりました。

“readme.txt”には，ファイルの概要が書かれていますので，参照してください。



《第4図》登場する敵キャラクターのタイプ数が増えても，この処理の応用でプログラミングしていけばいい



《第5図》敵キャラクターの固有部分の量で，処理スタイルをきめていこう

めて，方式を選択することが大切です。

敵キャラクター種類フラグと場合わけを考えよう

①の完全分離方式の場合，それぞれのキャラクター用にプログラム，変数とも用意することになるため，プログラミング技術としてはそれほどポイントになる点はありません。②の共通化方式の場合，一つのプログラムで2種類のキャラクターを処理することが必要となります。

このような処理の場合は，敵キャラクター用のデータに，敵キャラクターの種類を指定する“フラグ”を追加します。たとえば，A，B，Cの3タイプのキャラクターに対応する場合，etという敵キャラクターのタイプを指定するフラグ変数（配列変数）を用意します。このetが0であればAタイプの処理を，同様に1であればB，2であればCの処理を行います。こうすることで，一つの処理プログラムで三つのタイプのキャラクターに対応できるわけです（**第4図**）。

キャラクター間のちがいがそれほどない場合は，そのちがいがある部分だけ，このような場合わけを行えば済みます。他の処理は共通化できるので，プログラムが簡略化されるわけです。

逆に，キャラクターのちがいが大きい場合は，場合わけを行う部分が多くなってしまい，かえってプログラムが複雑になってしまいます。よって，共通化方式を利用する場合は，キャラクターの特徴のちがいが何項目，どの程度存在するかに注意する必要があります。

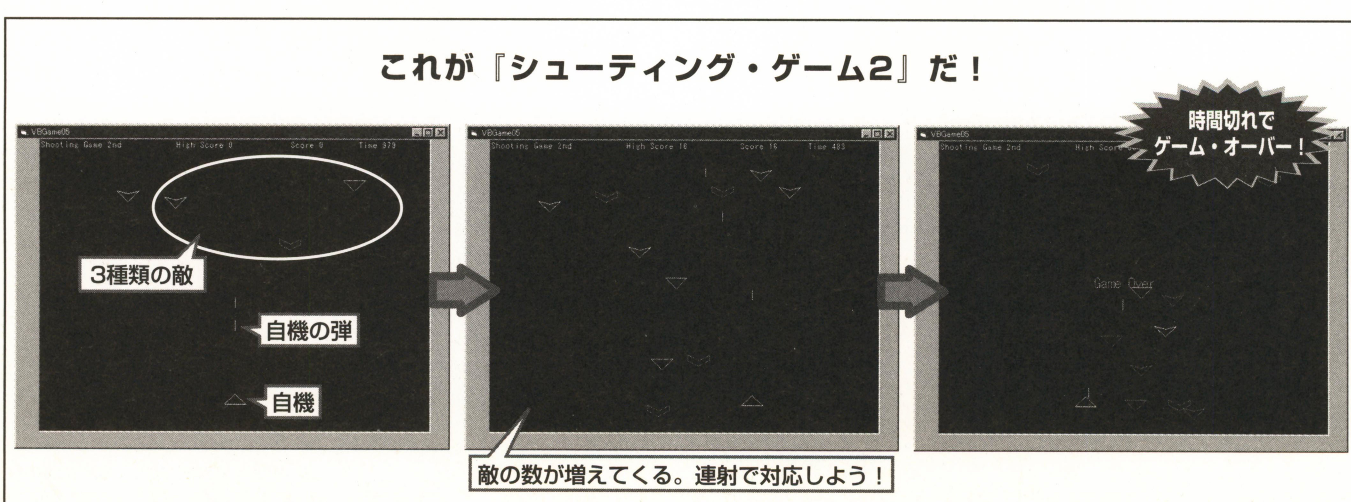
このことから，比較的ちがいが少ない雑魚キャラクター系は共通化で処理し，個性が重要なボス・キャラクター系は分離方式で処理するのが一般的です（**第5図**）。

『シューティング・ゲーム2』で体感してみよう！

それでは，実際にVBでプログラムされたシューティング・ゲームを実行してみましょう。プログラムは，**リスト1**です。

リストを入力して実行すると，**第6図**のような画面が表示されます。画面には，プレイヤーが操作するプレイヤー・キャラクター（自機）が登場します。また，画面上部から敵キャラクターが降下してきます。

プレイヤーは，自機をテンキーの[4][6]キーで左右に移動させます。敵キャラク



《第6図》3パターンの敵キャラクターが登場し，それぞれのパターンによって移動アルゴリズムが異なる

ターの体当たり攻撃を避け，スペース・キーで弾を発射して敵キャラクターを撃破しなくてはなりません。

弾が敵キャラクターに命中すれば，敵キャラクターは撃破されて得点がアップします。敵キャラクターが自機に接触すると自機はダメージを受け，残りプレイ時間が減少します（赤色の撃破した敵キャラクターに接触してもダメージを受けない）。プレイ時間がなくなるとゲーム・オーバーとなります。[Esc]キーでゲーム終了となります。

どんな敵キャラクターが登場する？

このゲームでは，敵キャラクターを第7図のように3とおり用意してあります。

敵キャラクターの処理プログラムは一つだけですが，変数etに敵キャラクターの種類を指定することで，表示処理や移動処理の段階で敵キャラクターの種類に応じた処理を実行するようにしています。

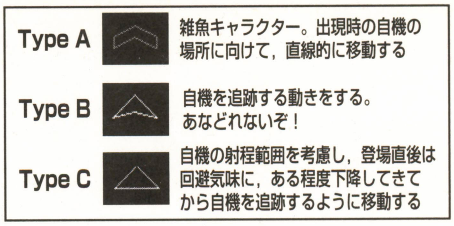
敵キャラクターの種類は，配列変数etに格納されています。プレイヤー・キャラクターとの衝突判定や発生処理など，共通している処理はすべての敵キャラクターについて実行します。敵キャラクターごとに異なる処理については，このetをif文でチェックし，etの内容に応じて分岐することで異なる処理を実行しています。

まとめ

敵キャラクターにバリエーションを持たせるということは，ゲームの構成に深みを持たせる効果があります。

そのときには，キャラクターごとの処理の内容を比較して，「個別プログラムを用意するのか」「プログラムは共通化してプログラム内部で分岐処理を行うか」ということをきめるといいでしょう。ぜひ，参考にしてください。

それでは，また来月をお楽しみに！



《第7図》3パターンの敵の特性を考えて，敵キャラクターの移動コースを先読みするようにしよう

《第1表》『シューティング・ゲーム2』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	プレイヤー・キャラクターX座標
py	プレイヤー・キャラクターY座標
ef()	敵キャラクター状態フラグ
ex()	敵キャラクターX座標
ey()	敵キャラクターY座標
evx()	敵キャラクターのX方向速度
evy()	敵キャラクター降下速度
et()	敵キャラクター・タイプ（0，1，2の3とおり）
pbf()	プレイヤーの弾の状態フラグ
pbx()	プレイヤーの弾のX座標
pby()	プレイヤーの弾のY座標
firereq	弾の発生要求フラグ
trig0flag	発射キーを押しているかどうかのフラグ
ti	残り時間
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

《リスト1》『シューティング・ゲーム2』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー
用キーの状態を記録する変数
Dim px!, py! 'プレイヤーキャラクタの座標
Dim ex!(9), ey!(9), ef%(9), et%(9), evx!(9), evy!(9) '敵
キャラクタ
Dim pbx!(2), pby!(2), pbf!(2) 'プレイヤーの弾
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim damage% 'ダメージ回数
Dim gamestat% 'ゲーム進行状況
Dim trig0flag% 'トリガキー状態
Dim firereq% 'プレイヤーの弾の発射要求フラグ
Dim col!(2) '敵キャラクタのカラー

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, i%
    With Form1
         .Width = Screen.TwipsPerPixelX * 640 'フォームの
サイズを設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame05"
    End With
    With Picture1
         .Width = Screen.TwipsPerPixelX * 576 'ピクチャボ
ックスのサイズを設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show
    col(0) = RGB(0, 255, 255)
    col(1) = RGB(255, 0, 255)
    col(2) = RGB(255, 255, 0)

    Picture1.Print "Shooting Game 2nd"

    px = 576 / 2 'プレイヤーキャラクタの初期座標
    py = 416 - 32

    For i = 0 To 9
        ef(i) = 0
    Next

    For i = 0 To 2
        pbf(i) = 0
    Next

    score = 0
    ti = 1000
    putscore

    trig0flag = 0

     gamestat = 0 'ゲームモード（0でゲーム中、1でゲーム
オーバー状態）

    Timer1.Interval = 30 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
     If (button And 2) = 2 Then 'マウス右ボタンクリックで
強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx!, i%, j%
    Dim npby!
    Dim nex!, ney!, dx!
    Dim x!
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        npx = px
         If keystat(4) = 1 Then 'キー状況に応じて新規座標
を変更
            npx = px - 5
        End If
        If keystat(6) = 1 Then
            npx = px + 5
        End If
        If npx < 20 Then npx = 20 '移動範囲制限
        If npx > 556 Then npx = 556
        putplayer px, py, 0 '元の場所のキャラクタを消去
         putplayer npx, py, RGB(255, 255, 255) '新しい場所
にキャラクタを描画
        px = npx 'プレイヤーキャラクタの座標を更新
        If keytrig0 = 1 Then '発射キーを押しているか？
            If trig0flag = 0 Then '一旦発射キーを離してい
るか？
                firereq = 1 'プレイヤーの弾の発生要求
                trig0flag = 1
            End If
        Else
            trig0flag = 0
        End If
        'プレイヤーの弾の処理
        For i = 0 To 2
            If pbf(i) = 0 Then
                 '存在していない場合は発射するかどうかを判
断
                If firereq = 1 Then
                    pbf(i) = 1 'プレイヤーの弾の発生
```

つぎのページに続く

リスト1の続き

```
                pbx(i) = px 'プレイヤーキャラクタの位置に発生
                pby(i) = py
                firereq = 0
            End If
        Else
            '存在している場合の処理
            npby = pby(i) - 13 '上昇
            If npby < 16 Then
                pbf(i) = 0 '画面上まで来たら消滅
                putpb pbx(i), pby(i), 0
            Else
                 For j = 0 To 9 '敵キャラクタとの衝突判定
                    If ef(j) = 1 And ex(j) + 16 > pbx(i) And ex(j) - 16 < pbx(i) And ey(j) + 16 > npby And ey(j) - 16 < npby Then
                        ef(j) = 2
                        scoreup 1
                        pbf(i) = 0
                    End If
                Next
                 putpb pbx(i), pby(i), 0 '元の位置から消去
                 If pbf(i) = 1 Then '存在している時だけ新しい場所に表示
                    putpb pbx(i), npby, RGB(255, 255, 0)
                End If
                pby(i) = npby '座標更新
            End If
        End If
    Next
    '敵キャラクタの処理
    For i = 0 To 9
        If ef(i) = 0 And score / 3 + 3 > i Then
            '敵キャラクタ発生処理
            ef(i) = 1
             ex(i) = Rnd * 500 + 38 '乱数でX座標を決定
            ey(i) = 32 '画面上部から登場させる
            et(i) = Int(Rnd * 3)    '乱数で敵キャラクタのパターンを決定
             evy(i) = Rnd * 3 + 3 '下降速度は乱数で調整
            If et(i) = 0 Then 'Type A
                 'Type Aの場合は発生した時の自機の位置にまっすぐ移動させる
                 evx(i) = (px - ex(i)) / (py - ey(i)) * evy(i)
            ElseIf et(i) = 1 Then 'Type B
                 'Type Bの場合は最初のX方向の速度は乱数
                evx(i) = Rnd * 5 - 2
             Else 'Type Cの場合は最初のX方向の速度は自機を避ける方向
                evx(i) = Rnd * 3
                If px > ex(i) Then evx(i) = -evx(i)
            End If
        ElseIf ef(i) = 1 Then
            '敵キャラクタ移動処理
            ney = ey(i) + evy(i) '降下
            If et(i) = 0 Then 'Type A
                 nex = ex(i) + evx(i) '登場時の設定で直進
            ElseIf et(i) = 2 Then 'Type B
                 x = ex(i) + evx(i) * (py - ey(i)) / evy(i)   '自機の位置まで降下してきた時のX座標を推定
                 If x > px + 8 Then evx(i) = evx(i) - 0.5 '自機に衝突するようにX方向速度を補正
                 If x < px - 8 Then evx(i) = evx(i) + 0.5
                If evx(i) > 5 Then evx(i) = 5
                If evx(i) < -5 Then evx(i) = -5
                nex = ex(i) + evx(i)
            Else 'Type C
                If ney < 240 Then '画面上部に存在するか？
                    'プレイヤーキャラクタを避ける動き
                    If ex(i) > px Then
                        dx = 100 / (ex(i) - px + 10)
                        If dx > 3 Then dx = 3
                        nex = ex(i) + dx
                        If nex > 556 Then nex = 556
                    Else
                        dx = 100 / (px - ex(i) + 10)
                        If dx > 3 Then dx = 3
                        nex = ex(i) - dx
                        If nex < 20 Then nex = 20
                    End If
                Else
                    'プレイヤーキャラクタに突っ込む動き
                    If ex(i) < px Then
                        nex = ex(i) + 3
                        If nex > 556 Then nex = 556
                    Else
                        nex = ex(i) - 3
                        If nex < 20 Then nex = 20
                    End If
                End If
            End If
             If ney > py - 8 And ney < py + 8 And nex > px - 16 And nex < px + 16 Then
                ti = ti - 100 'プレイヤーキャラクタに衝突
                ef(i) = 0
            End If
            If ney > 416 Then ef(i) = 0
            putenemy ex(i), ey(i), et(i), 0
             If ef(i) = 1 Then putenemy nex, ney, et(i), col(et(i))
            ex(i) = nex '座標更新
            ey(i) = ney
        ElseIf ef(i) = 2 Then
            '敵キャラクタやられ処理
            ney = ey(i) + evy(i)
              putenemy ex(i), ey(i), et(i), RGB(0, 0, 0)
            If ney < 416 Then
                 putenemy ex(i), ney, et(i), RGB(255, 0, 0)
            Else
                ef(i) = 0 '消滅
            End If
            ey(i) = ney '座標更新
        End If
    Next
    ti = ti - 1
    If ti < 0 Then ti = 0
    puttime
     If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
ElseIf gamestat = 1 Then
    With Picture1
        .CurrentX = 230
        .CurrentY = 200
        .FontSize = 15
         .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
    End With
    Picture1.Print "Game Over"
    If keytrig3 = 1 Then
        Timer1.Interval = 0 'インターバルタイマ停止
        gameinit
    End If
End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Line (x, y - 8)-(x + 16, y + 8), c
    Picture1.Line (x, y - 8)-(x - 16, y + 8), c
    Picture1.Line (x + 16, y + 8)-(x - 16, y + 8), c
End Sub

Private Sub putpb(x, y, c)
    Picture1.Line (x, y - 8)-(x, y + 8), c
End Sub

Private Sub putenemy(x, y, t, c)
    If t = 0 Then
        Picture1.Line (x + 16, y - 8)-(x - 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y + 8), c
        Picture1.Line (x - 16, y - 8)-(x, y + 8), c
    ElseIf t = 1 Then
        Picture1.Line (x + 16, y)-(x + 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y), c
        Picture1.Line (x, y)-(x - 16, y - 8), c
        Picture1.Line (x - 16, y - 8)-(x - 16, y), c
        Picture1.Line (x - 16, y)-(x, y + 8), c
        Picture1.Line (x, y + 8)-(x + 16, y), c
    Else
        Picture1.Line (x, y + 8)-(x + 16, y - 8), c
        Picture1.Line (x + 16, y - 8)-(x, y - 4), c
        Picture1.Line (x, y - 4)-(x - 16, y - 8), c
        Picture1.Line (x - 16, y - 8)-(x, y + 8), c
    End If
End Sub
```

リスト1 入力&実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，「標準EXE」を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真**①のような画面が表示されます。

②フォームの設定

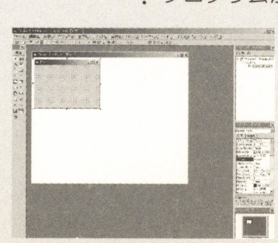
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意)。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される)。

③リストの入力

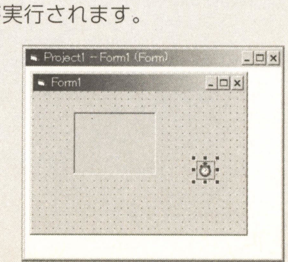
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，リスト1を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます。



《写真①》VBを起動したら「標準EXE」を選ぼう



《写真②》PictureBoxとTimerをフォームに配置する

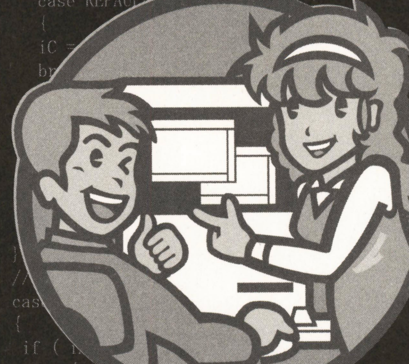
プログラマー&クリエイターへの登竜門

BM特選 投稿プログラムコーナー

- Visual Basic
- DirectX for Visual Basic
- Excel VBA
- Visual C++
- F-BASIC
- HSP
- Q Basic
- iアプリ
- ワンダーウィッチ
- PC-E650
- MUSIC

本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG” というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1）プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注2）プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3）ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

★ここに掲載されているプログラムは，ベーマガ・ホームページからもダウンロードできます。また，ホームページからの投稿も受け付けています。

http://www.basicmagazine.net/

Windows(Visual Basic V6.0)用



ハンドリング操作がすべてだ

THE（ザ） カーチェイス

日下　心平（15歳）

ストーリー

ここは高速を気持ちよく走っている刑事の車の中。突然電話が入る。

「警部！今，犯人から犯行声明文が送られてきました！」

「何と書いているんだ？」

「警部の車に爆弾がセットされているようです」

「ぐふっ何！すぐ車から出なければ…」

「ちなみに車を止めると爆発するそうです」

「げっ…なんとかならないのか？」

「一定以上の速さである程度走れば，爆弾は解除されると書いています。」

「一定以上ってどのくらい？」

「少なくとも時速160kmくらいだと書いています。」

「なんかやけにていねいな犯行声明文だな…なんでそんなことをわざわざ書くんだろう？」

「じゃあがんばってくださいね！」

電話が切れる。

「おいおい，これじゃ生きて帰っても免許取り上げじゃないか！」

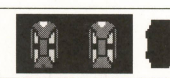
こんなことを言っても内心めちゃくちゃなスピードで走れることを喜ぶ警部であった。

内容・使いかた・遊びかた

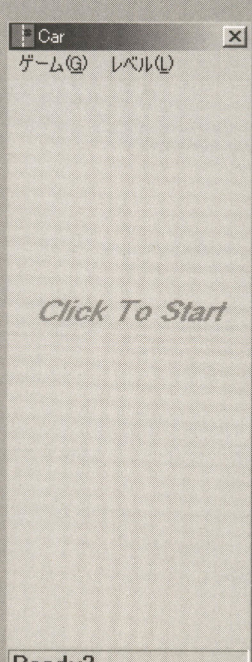
ただ前方の車を避けるだけのシンプルなよけゲーです。

プログラムを起動し，レベルメニューからレベルを選択してからウインドウをクリックしてスタートしてください。赤い車が自分，青い車が敵です。

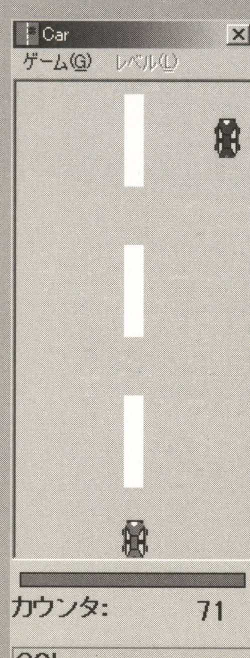
レベルには六つあります。1〜5では，速さが違います。Night Modeでは，暗



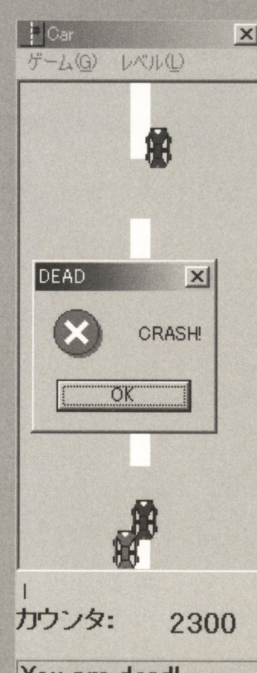
car.bmpの画像



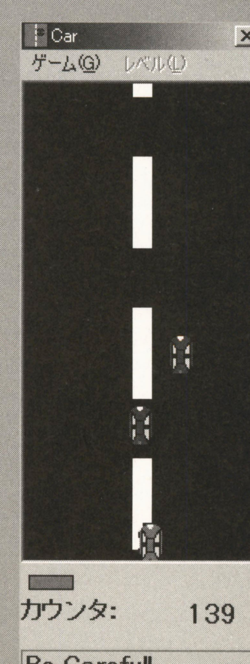
《写真1》レベルを設定をしたらゲーム・スタート



《写真2》向かってくる車を避けるのだ



《写真3》先を見て操作しないとクリアは難しい



《写真4》見とおしの悪いNightModeにも挑戦だ

《OVER FLOW》夏休み，あー夏休み，夏休み。というわけで，いい汗をかき，そして，いろんなことを吸収・勉強する絶好の時期です。思い出をたくさん作ってください。ベーマガ読者ならもちろん，プログラム作品作りも忘れないように！投稿お待ちしています。

CHECKER FLAG

影： カーチェイスというタイトルだけど，内容としては障害避けみたいな感じ。ホントに初期のカーレースを連想させるゲーム内容だ。

Dr.D： 1年でここまで作れればなかなかのものじゃろう。これからも自分のペースでプログラムの勉強を続けることが大切じゃな。

つぐ美： 白熱したゲームにしたいなら，まずは映画などでいろいろな車のシーンを見るのがオススメ。今回みたいなゲームを作りたいなら「スピード」という映画は1度は見ておきましょう。あとは，映画ではないけど「西部警察」のカーチェイス場面もかなりオススメね。

編： プログラムなどの専門書は確かに値段が高めですね。購入するときは本屋でしっかり吟味するのが大切でしょう。あとは図書館などを利用するという手があります。

くて見とおしが悪く，敵が途中で出現。カウンタが2000になると速くなります。カウンタが4000以上になるとクリアです。

ゲーム中にゲームメニューのリセットを押すとタイトルに戻ります。

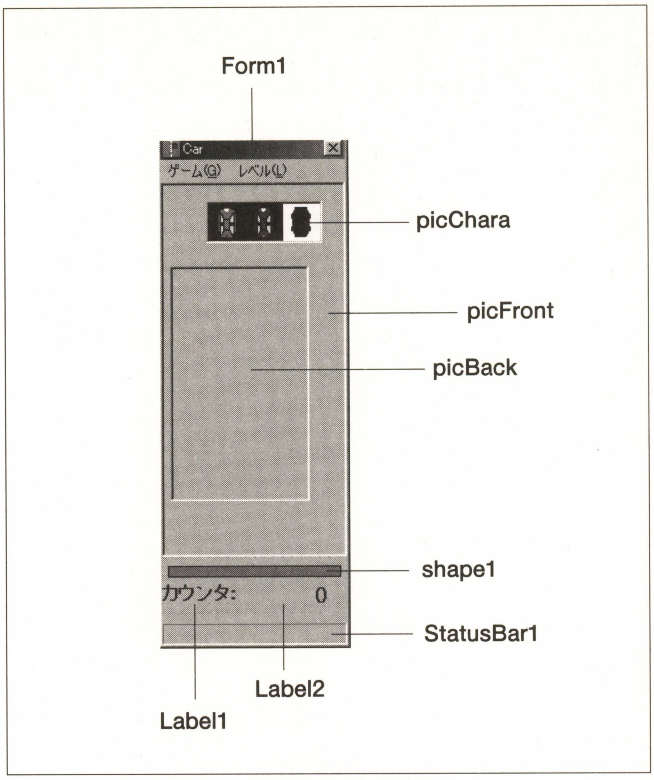
操作方法は，カーソル・キーの上下左右で動かします。

《第1表》変数表

start	ゲームは始まっているか
rightkey	右キーが押されているか
leftkey	左キーが押されているか
upkey	上キーが押されているか
downkey	下キーが押されているか
jiki	自機
teki(5)	敵機
hp	自機のＨＰ
getpoint	カウンタ
road	道路の状態
nightmode	ナイトモードか
i	ループカウンタ
siten	道路描画の始点
syutugen	敵の出現位置
backcol	バックカラー

プログラムの入力法と注意点

プロジェクトメニューのコンポーネントでMicrosoft Windows Common Control 6.0にチェックを付けておいてください。それからFor文用のループカウンタiは何度も出てきていますが，これはコードを見やすくするためのものです(たまに自分でもわからなくなるから)。メモリがもったいないという人は一つにまとめるのがいいでしょう。



フォームの配置図

改造のアドバイスと苦労点

最低限の機能しかつけてないので改造の余地はいくらでもあります（たとえばクリア条件など)。

もうちょっと白熱できるゲームにしたい。あとは，名前があまりにありきたりなのが反省点です。

ひとこと

初めてグラフィカルなゲームが出できました。VB買ってから1年近くになります。これは遅いほうでしょうか？プログラミングの本を買うとかなりの量のお金を消費します。どうしましょう？

《第2表》メニューの設定

ｷｬﾌﾟｼｮﾝ(P)	名前(M)	ｲﾝﾃﾞｯｸｽ(X)	ﾍﾙﾌﾟｺﾝﾃｷｽﾄ番号(H)	ｼｮｰﾄｶｯﾄ(S)	ﾈｺﾞｼｴｰﾄ位置(O)	ﾁｪｯｸ(C)	有効(E)	表示(V)	ｳｲﾝﾄﾞｳ ﾘｽﾄ(W)
ゲーム(&G)	mnugame		0	(なし)	0 －なし		○	○	
…リセット(&R)	mnuReset		0	(なし)	0 －なし		○	○	
…終了(&X)	mnuExit		0	(なし)	0 －なし		○	○	
レベル(&L)	mnuLevel		0	(なし)	0 －なし		○	○	
…Level 1	mnuL	1	0	(なし)	0 －なし		○	○	
…Level 2	mnuL	2	0	(なし)	0 －なし		○	○	
…Level 3	mnuL	3	0	(なし)	0 －なし	○	○	○	
…Level 4	mnuL	4	0	(なし)	0 －なし		○	○	
…Level 5	mnuL	5	0	(なし)	0 －なし		○	○	
…Night Mode	mnuL	6	0	(なし)	0 －なし		○	○	

《第3表》プロパティ表

フォームの設定	
Name	Form1
AutoRedraw	-1 'True
BorderStyle	1 '固定(実線)
Caption	Car
Font	MS Sans Serif
ForeColor	&H00FF00FF&
Height	6600
Icon	(アイコン)
KeyPreview	-1 'True
Left	4965
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	396
ScaleMode	3 'ピクセル
ScaleWidth	160
Top	2955
Width	2490
ピクチャボックスの設定	
Name	picChara
AutoRedraw	-1 'True
AutoSize	-1 'True
Font	ＭＳ Ｐゴシック
Height	36
Left	40
Picture	car.bmp
ScaleHeight	32
ScaleMode	3 'ピクセル
ScaleWidth	96
TabIndex	2
Top	16
Visible	0 'False
Width	100
Name	picBack
AutoRedraw	-1 'True
Font	ＭＳ Ｐゴシック
Height	200
Left	8
ScaleHeight	196
ScaleMode	3 'ピクセル
ScaleWidth	116
TabIndex	1
Top	72
Visible	0 'False
Width	120
Name	picFront
AutoRedraw	-1 'True
Font	ＭＳ Ｐゴシック
Height	320
Left	0
ScaleHeight	316
ScaleMode	3 'ピクセル
ScaleWidth	156
TabIndex	0
Top	0
Width	160
ラベルの設定	
Name	Label1
AutoSize	-1 'True
Caption	カウンタ：
Font	ＭＳ Ｐゴシック
Height	16
Left	0
TabIndex	3
Top	344
Visible	0 'False
Width	68
Name	Label2
Alignment	1 '右揃え
Caption	0
Font	ＭＳ Ｐゴシック
Height	17
Left	64
TabIndex	4
Top	344
Visible	0 'False
Width	81
シェイプの設定	
Name	Shape1
FillColor	&H000000FF&
FillStyle	0 '塗りつぶし
Height	10
Left	5
Top	328
Visible	0 'False
Width	150
ステータスバーの設定	
Name	StatusBar1
Align	2 '下揃え
Font	ＭＳ Ｐゴシック
Height	20
Left	0
Style	1
TabIndex	5
Top	376
Width	160

★うわぁーっ！あんなに苦労して作ったゲームがボツるなんて！あのゲームが載ってくれることを願ってます。私の最後になるかもしれないあのゲーム，載ってくれぃ！（東京都江東区・**マーサン15歳**）……【**編**：今月号に載りましたね。最後と言わず，よろしく！】

『THEカーチェイス』のプログラム・リスト

```
Option Explicit
'BitBltを宣言//////////////////////////////////////////////////////
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As _
Long, ByVal x As Long, ByVal y As Long, ByVal nWidth As Long, _
ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long, _
ByVal ySrc As Long, ByVal dwRop As Long) As Long
'Sleepを宣言///////////////////////////////////////////////////////
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Dim start As Boolean 'ゲームは始まっているか
Dim rightkey As Boolean '右キーが押されているか
Dim leftkey As Boolean '左キーが押されているか
Dim upkey As Boolean '上キーが押されているか
Dim downkey As Boolean '下キーが押されているか
Dim jiki As chara '自機
Dim teki(5) As chara '敵機
Dim hp As Byte '自機のＨＰ
Dim getpoint As Long 'カウンタ
Dim road As Byte '道路の状態
Dim nightmode As Boolean 'ナイトモードか

Private Type chara '構造体
    x As Integer 'X座標
    y As Integer 'Y座標(Warning!敵のY座標はピクチャボックスの一番下が0で自機と逆)
    Flag As Boolean '生きてるか(敵機に対しては使っていません。無駄かも。)
    v As Integer '速度(自機に対しては使っていません。一ループに対する移動距離)
End Type


Private Sub Form_Click()
    Dim i As Byte 'ループカウンタ
    If start = False Then
        start = True
        picFront.Visible = True
        jiki.Flag = True
        For i = 0 To 5 '敵機の初期設定↓↓↓//////////////////////////////////////↓↓↓
            teki(i).Flag = True
            teki(i).x = random(0, 128) 'X座標を設定/////////////////////////////
            teki(i).y = random(800, 1600) 'Y座標を設定///////////////////////////
            If mnuL(1).Checked = True Then teki(i).v = 6 '速度を設定↓↓↓///↓↓↓
            If mnuL(2).Checked = True Then teki(i).v = 7
            If mnuL(3).Checked = True Then teki(i).v = 8
            If mnuL(4).Checked = True Then teki(i).v = 9
            If mnuL(5).Checked = True Then teki(i).v = 11
            If mnuL(6).Checked = True Then teki(i).v = 10: nightmode = True
                                                      '速度を設定↑↑↑///↑↑↑
        Next '敵機の初期設定↑↑↑//////////////////////////////////////////↑↑↑
        picBack.Line (0, 0)-(picBack.Width, picBack.Height), &H8000000F, BF
        Label1.Visible = True: Label2.Visible = True: Shape1.Visible = True
        StatusBar1.SimpleText = "GO!"
        CharaBlt
        mnuLevel.Enabled = False
        game 'ゲームスタート//////////////////////////////////////////////////
    End If
End Sub


'キー入力判定↓↓↓////////////////////////////////////////////////////////↓↓↓
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyLeft Then leftkey = True
    If KeyCode = vbKeyRight Then rightkey = True
    If KeyCode = vbKeyUp Then upkey = True
    If KeyCode = vbKeyDown Then downkey = True
End Sub


Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyLeft Then leftkey = False
    If KeyCode = vbKeyRight Then rightkey = False
    If KeyCode = vbKeyUp Then upkey = False
    If KeyCode = vbKeyDown Then downkey = False
End Sub
'キー入力判定↑↑↑////////////////////////////////////////////////////////↑↑↑


Private Sub Form_Load()
    Label1.BackColor = RGB(25, 240, 200)
    Label2.BackColor = RGB(25, 240, 200)
    shokika '初期化
End Sub
```


★ページをもっと増やしてほしい。Javaプログラムのことを載せてほしい。（新潟県長岡市・南　寿和19歳）……【編：ちなみに，愛読者ハガキでアンケートをとっている「これから勉強したいプログラム言語」の第3位にはJavaが入っていますね。さて1位は？】

```
Sub shokika() '初期化////////////////////////////////////////////////////////
    Randomize Time
    start = False
    mnuLevel.Enabled = True
    picFront.Height = 320: picFront.Width = 160 'スクリーンの大きさ
    picBack.Height = 356: picBack.Width = 160
    picFront.Visible = False
    nightmode = False
    jiki.x = 64: jiki.y = 320 '自機の座標を初期化
    hp = 10 'ＨＰを初期化
    getpoint = 0 'カウンタを初期化
    Show 'フォームを表示
    BackColor = RGB(25, 240, 200)
    leftkey = False: rightkey = False: downkey = False: upkey = False
    Label2.Caption = "0": Shape1.Width = 150 '各コントロールを初期化
    Label1.Visible = False: Label2.Visible = False: Shape1.Visible = False
    StatusBar1.SimpleText = "Ready?"
    road = 0 '道路の状態を初期化
    CurrentX = 15: CurrentY = 140: Print "Click To Start"

End Sub


Sub CharaBlt()
    Dim i As Byte 'ループカウンタ
    Dim siten As Byte '道路描画の始点
    Dim syutugen As Integer '敵の出現位置
    Dim backcol As Long 'バックカラー
    'ナイトモードだったら
    If nightmode = False Then
    syutugen = 320: backcol = &H8000000F
    Else
    syutugen = 260: backcol = &H8000008F
    End If
    '道路の線を描画////////////////////////////////////////////////////////////
    If road <= 5 Then siten = 0 Else siten = road * 10 - 60
    picBack.Line (72, siten)-(84, road * 10), vbWhite, BF
    picBack.Line (72, road * 10 + 40)-(84, road * 10 + 100), vbWhite, BF
    picBack.Line (72, road * 10 + 140)-(84, road * 10 + 200), vbWhite, BF
    picBack.Line (72, road * 10 + 240)-(84, road * 10 + 300), vbWhite, BF

    '自分の描画/////////////////////////////////////////////////////////////////
    BitBlt picBack.hDC, jiki.x, jiki.y, 32, 32, picChara.hDC, 64, 0, vbSrcAnd
    BitBlt picBack.hDC, jiki.x, jiki.y, 32, 32, picChara.hDC, 0, 0, vbSrcInvert
    '敵の描画///////////////////////////////////////////////////////////////////
    For i = 0 To 5
    If teki(i).Flag = True Then
        If teki(i).y <= syutugen Then '表示できる位置にある場合/////////////////////
        BitBlt picBack.hDC, teki(i).x, 320 - teki(i).y, 32, 32, picChara.hDC, _
        64, 0, vbSrcAnd
        BitBlt picBack.hDC, teki(i).x, 320 - teki(i).y, 32, 32, picChara.hDC, _
        32, 0, vbSrcInvert
    End If: End If
    Next i
    'オフスクリーンから表に//////////////////////////////////////////////////////
    BitBlt picFront.hDC, 0, 0, 160, 320, picBack.hDC, 0, 32, vbSrcCopy
    picFront.Refresh
    'オフスクリーンを消去/////////////////////////////////////////////////////////
    picBack.Line (0, 0)-(picBack.Width, picBack.Height), backcol, BF
End Sub


Private Sub game() 'メインループ/////////////////////////////////////////////
If start = True Then
Do Until (jiki.Flag = False) Or getpoint >= 4000 '死ぬかカウンタが4000以上になるまでループ
    If start = False Then GoTo resetout 'リセットボタンがクリックされた///
    getpoint = getpoint + 1 'カウンタを加算//////////////////////////////////////
    Label2.Caption = Str(getpoint) 'カウンタを表示/////////////////////////////
    CharaChange '座標更新
    Shape1.Width = hp * 15 'HPのバーを更新
    If hp = 0 Then jiki.Flag = False '死に/////////////////////////////////////////
    CharaBlt '描画
    DoEvents
    'Wait20ミリ秒///////////////////////////////////////////////////////////
    Sleep 20
Loop
    If jiki.Flag = False Then 'ゲームオーバー///////////////////////////
        StatusBar1.SimpleText = "You are dead!"
        MsgBox "CRASH!", vbCritical, "DEAD"
```

★6月号の「バンダイロボット研究所」の記事がおもしろかったです。もっとページがほしかったとこですが。（和歌山県西牟?郡・ねこまんま18歳）……**【影：エレクトロニック研究所訪問記は人気上昇中！訪問してもらいたい機関などあったら，お便りください。】**

```
        Else 'ステージクリア/////////////////////////////////////////////
            If hp <> 10 Then
                StatusBar1.SimpleText = "Stage Clear!"
            Else
                StatusBar1.SimpleText = "Parfect Play!"
            End If
            MsgBox "You are alive! Congratulations!", vbOKOnly + vbInformation, "Clear"
        End If
        shokika
    resetout:

    End If
    End Sub


    Private Sub Form_Unload(Cancel As Integer)
        End
    End Sub


    '乱数発生関数(min,乱数の最小値 max,乱数の最大値)/////////////////////////
    Private Function random(min As Integer, max As Integer) As Integer
        random = Int(min + (Rnd * (max - min)))
    End Function


    Private Sub CharaChange() 'キャラの座標更新///////////////////////////
    Dim i As Byte
    '自機の座標更新//////////////////////////////////////////////////////
    If jiki.x > 0 Then
        If leftkey = True Then jiki.x = jiki.x - 5
    End If
    If jiki.x < 128 Then
        If rightkey = True Then jiki.x = jiki.x + 5
    End If
    If jiki.y > 32 Then
        If upkey = True Then jiki.y = jiki.y - 5
    End If
    If jiki.y < 320 Then
        If downkey = True Then jiki.y = jiki.y + 5
    End If
    '敵機の座標更新(あたり判定付！) ////////////////////////////////////////
    For i = 0 To 5
        If getpoint = 2000 Then teki(i).v = teki(i).v + 1 'カウンタが2000になったら速くする
        teki(i).y = teki(i).y - teki(i).v '敵機の座標更新/////////////////
        '敵が一番下まできたら上に戻す////////////////////////////////////////
        If teki(i).y < -32 Then
        teki(i).y = random(320, 720)
        teki(i).x = random(0, 128)
        End If
        '当たり判定//////////////////////////////////////////////////////
        If Abs(jiki.y - (320 - teki(i).y)) < 24 And Abs(jiki.x - teki(i).x) < 15 Then
            hp = hp - 1
            getpoint = Int(getpoint * 9.5 / 10) 'カウンタを減点
            If hp = 0 Then Exit Sub
        End If
    Next
        If road = 8 Then road = 0 Else road = road + 1 '道路の状態を更新
        If hp <= 3 Then StatusBar1.SimpleText = "Be Careful!"
    End Sub

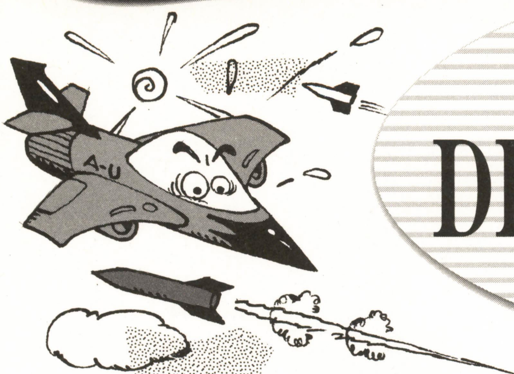

    Private Sub mnuExit_Click() '終了
        Unload Me
    End Sub


    Private Sub mnuL_Click(Index As Integer) 'レベル選択
        Dim i As Byte
        'メニュー(レベル) のチェックの表示・非表示////////////////////
        For i = 1 To 6
        If mnuL(i).Checked = True Then mnuL(i).Checked = False
        Next
        mnuL(Index).Checked = True
    End Sub


    Private Sub mnuReset_Click() 'ゲームをリセット(初期画面に戻す)
        shokika '初期化
    End Sub
```


★「ラジオの製作」が休刊となってから，ベーマガに移ってきました。社会人にも内容が楽しめ，プログラムの勉強になります。できれば，VBは卒業して，VC++の記事を増やしてくれればと思います。(栃木県宇都宮市・**ザマミマル41歳**)……**【編：貴重なご意見です。】**

Windows(Visual Basic V6.0)用



5方向ショットとボムを使いこなせ

DESTRUCTION GOD

（デストラクション ゴッド）

Shota（13歳）

内容

弾よけシューティングです。敵の弾をとにかくよけてください。一定時間で難易度がどんどん上がっていきます。ゲームクリアはありません。自分の弾よけ能力の限界に挑戦してください。

ストーリー

おれの愛機「DESTRUCTION GOD」は世界最強の戦闘機だ。広範囲攻撃可能な高速連射弾を装備している。ただし弱点もある。それはおれが趣味で作ったものだけにもろいことだ。

ある日，おれは我が愛機に乗って宇宙を旅した。そこで事件はおきた。はるか彼方から群れをなして飛んでくる無数の戦闘機。どうやらおれは彼らの領域に入ってしまったらしい。

世界最強の戦闘機に敵はいない！！！絶対，ここを通り抜けてみせるぜ。

使いかた・遊びかた

↑↓←→キーで自機が移動します。Zキーでショットを打ちます。自動連射です。Xキーでボムを発動できます。ボムは画面内の敵・敵弾を一気に消し去ります。ただし，使用回数に制限があります。

プログラムの仕組み

Scene変数にシーン（タイトル，ゲーム中）の番号を記録していて，メインループからシーン別の関数を呼び出しています。これがシューティングの基本的な構造だと思います。

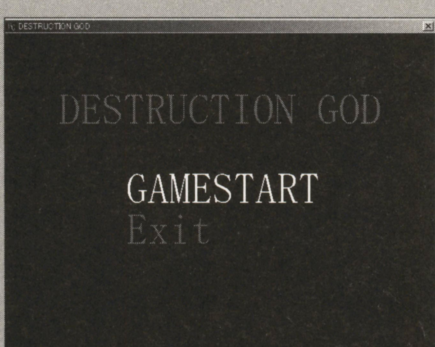
プログラムの入力方法と注意点

自機と敵のグラフィックを用意してください。自機，敵とも48×48の大きさで，自機，自機マスク，敵，敵マスクの順に並べてください。

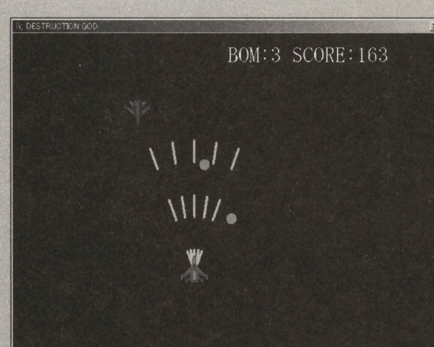
改造のアドバイス

容量の関係上グラフィックがVBの描画メソッドを使っていたりするので，ビットマップで奇麗なグラフィックを作ってBitBltで描画する。背景をつけるのは弾が見やすい範囲にとどめましょう。

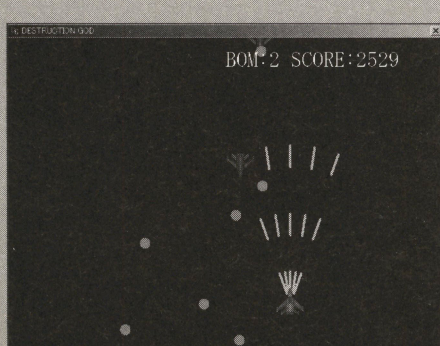
あとは，効果音，BGMをつける。敵の弾の種類を増やす(ホーミング弾，レーザー



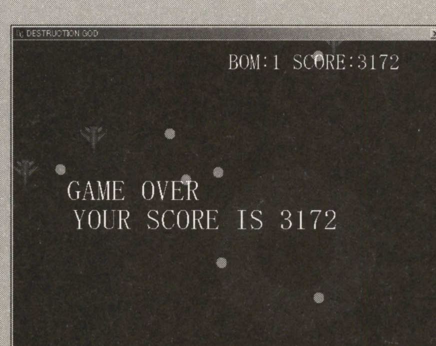
《写真1》基本的なシューティング・ゲーム



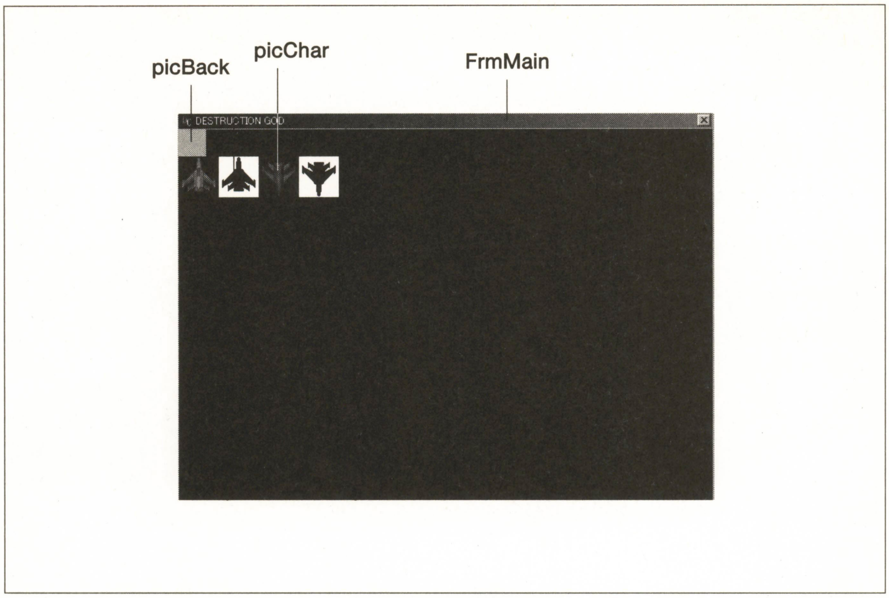
《写真2》5方向ショットを使いこなせ



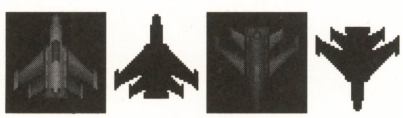
《写真3》進行にともなって敵も増えていくゾ



《写真4》ハイスコアを叩きだせるか？



フォーム配置図



char.bmpの画像

など)。

苦労したところ

ソースをコンパクトにまとめるところです。

作者からのコメント

コードは結構まじめに書いたので，見やすいと思います。最近小学生プログラマーからの投稿があるようで，僕も負けていられないので，一応お手本になるようなコードを書いたつもりです。書きかた次第ではまだ最適化の余地があるかもしれません。やっぱりシューティングは結構大きくなりますね。今後はもう少しコンパクトな作品を投稿してみたいと思います。

それから，原稿ファイルを書くのにベーマガ5月号の『Basic Magazine投稿しようよ』(谷澤友哉君作）を使わせていただきました。結構使い勝手がよかったと思います。今後の改良に期待します。

《第１表》プロパティ表

フォームの設定	
Name	frmMain
AutoRedraw	-1 'True
BackColor	&H00000000&
BorderStyle	1 '固定(実線)
Caption	DESTRUCTION GOD
ClipControls	0 'False
Height	7575
Icon	(アイコン)
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	480
ScaleMode	3 'ピクセル
ScaleWidth	640
StartUpPosition	3 'Windows の既定値
Top	0
Width	9690
ピクチャボックスの設定	
Name	picChar
AutoRedraw	-1 'True
AutoSize	-1 'True
BorderStyle	0 'なし
ClipControls	0 'False
Height	48
Left	0
Picture	char.bmp
ScaleHeight	720
ScaleWidth	2880
TabIndex	1
Top	32
Visible	0 'False
Width	192
Name	picBack
AutoRedraw	-1 'True
BorderStyle	0 'なし
ClipControls	0 'False
Font	ＭＳ 明朝
Height	33
Left	0
ScaleHeight	33
ScaleMode	3 'ピクセル
ScaleWidth	33
TabIndex	0
Top	0
Visible	0 'False
Width	33

CHECKER FLAG

Dr.D： ５方向分裂のショットや，ランダムの値から三角関数を使って敵の動きを作り出すなど，基本的なシューティングの処理が取り入れられている作品じゃ。気になるのはゲーム全体が単調になっている点じゃな。プログラム全体の構成はうまくできているので，ここからはゲーム内容を重視した改造を行っていってほしいものじゃ。また，三角関数に与える値でやたら長い小数があるな。こういう値は定数として定義して使用したほうが入力ミスも防げるし，見やすいリストになるゾ。

影： 今のベーマガのウリの一つは，Shotaクンみたいな若い世代からの投稿があることだね。中学生でここまでゲームプログラムが組める人が出てきてくれるのは喜ばしいことだ。

編： 『Basic Magazine投稿しようよ』は記入もれを防ぐことができるので，編集部からもオススメのツールです。ぜひ，ベーマガＨＰの投稿プログラムページからダウンロードしてご利用ください。

《リスト１》フォームのプログラム・リスト

```
'=========================================
'    DESTRUCTION GOD -メインウィンドウ-
'=========================================
Option Explicit

'============
'    初期化
'============
Private Sub Form_Load()
    On Error GoTo Error
    Show
    DoEvents
    'バックバッファ作成
    Call picBack.Move(0, 0, 640, 480)
    picBack.BackColor = vbBlack

    Randomize
    Call SystemInit
    Call timeBeginPeriod(1) 'タイマー精度設定
    Call MainLoop   'メインループ呼び出し
    Call timeEndPeriod(1)
Error:
    Unload Me
End Sub

'==========
'    終了
'==========
Private Sub Form_Unload(Cancel As Integer)
    On Error Resume Next
    End
End Sub
```

《第２表》変数表

i,j	ループカウンタ
Char型構造体	キャラクターを扱います
…Use	使用中かどうかのフラグ
…t	敵の種類識別用(0:敵，1:敵弾，2:爆発)
…x	キャラクターのx座標
…y	キャラクターのy座標
…vx	x方向の移動量
…vy	y方向の移動量
…Shield	敵の耐久力 自機の場合はゲーム・オーバーになってからの時間カウントに使用
…Frame	経過フレーム数
Jiki	自機キャラクター。Char型
Jshot(49)	自機の弾。Char型
Enemy(255)	敵，敵弾，爆発。Char型
Level	難易度
Bom	ボム使用可能回数
Score	得点
PressKey～	～の部分はキーの名前。その キーが押されているフレーム数
Frame	フレームカウンタ
MainLoopExit	メインループを抜けるフラグ
Scene	シーン番号
index	タイトル画面で選択している項目

★今月やってしまいました。まちがって2冊も買ってしまいました。そういえば，サッカーすごいですよね（コン杯)。まさかフランス（主力いなかったけど）に0−1とは，健闘したと思います。(静岡市・**多田和彦22歳**）……【**影：何がやってしまったのじゃ？あれれ…。**】

《リスト2》モジュール basGame.basのプログラム・リスト

```
'====================================
'   DESTRUCTION GOD  -ゲーム本体-
'====================================
Option Explicit
Private Type Char       'キャラクタ型
    Use As Boolean  'フラグ
    t As Long       '種類
    x As Single     '座標
    y As Single
    vx As Single        '移動量
    vy As Single
    Shield As Long  '耐久力
    Frame As Long   '経過フレーム数
End Type
Private Jiki As Char           '自機
Private Jshot(49) As Char      '自機弾
Private Enemy(255) As Char     '敵
Private Level As Long          '難易度
Private Bom As Long            'ボム
Private Score As Long          '得点

'================
'   ゲーム本体
'================
Public Sub GameMain()
    Dim i As Long
    If Frame = 0 Then
        '初期化
        With Jiki
            .x = 320
            .y = 320
            .Shield = 0
        End With
        For i = 0 To 49
            Jshot(i).Use = False
        Next i
        For i = 0 To 255
            Enemy(i).Use = False
        Next i
        Level = 1
        Bom = 3
        Score = 0
    End If
    '============
    '   敵出現
    '============
    If Frame Mod ((60 / Level) * 5) = 0 Then
        i = CLng(Rnd * 120) + 30
        Call CreateEnemy(0,Rnd*320+160,-24,Cos(i*0.017453292517)*3,Sin(i*0.017453292517)*3)
    End If
    '一定時間で難易度アップ
    If Frame Mod 300 = 0 And Level < 30 Then Level = Level + 1
    '==========
    '   移動
    '==========
    Call MoveEnemy
    If Jiki.Shield = 0 Then
        Call MoveJiki
        Call MoveJshot
    End If
    '==========
    '   描画
    '==========
    Call DrawEnemy
    If Jiki.Shield = 0 Then
        Call DrawJshot
        Call DrawJiki
    End If
    'スコア
    With frmMain.picBack
        .ForeColor = vbWhite
        .Font.Size = 24
        .CurrentX = 320
        .CurrentY = 16
    End With
    frmMain.picBack.Print ("BOM:" & CStr(Bom) & " SCORE:" & CStr(Score))
    '======================
    '   ゲームーオーバー
    '======================
    If Jiki.Shield < 0 Then
        Jiki.Shield = Jiki.Shield - 1
        If Jiki.Shield = -200 Then Call ChangeScene(SN_TITLE)
        With frmMain.picBack
            .CurrentX = 80
            .CurrentY = 200
            .FontSize = 32
            .ForeColor = vbWhite
        End With
        frmMain.picBack.Print ("GAME OVER" & vbCrLf & "    YOUR SCORE IS " & CStr(Score))
    Else
        '生きている
        Score = Score + 1
    End If
End Sub

'==========
'   自機
'==========
Private Sub MoveJiki()
    Dim i As Long
    Dim j As Long
    With Jiki
        If PressKeyUp Then .y = .y - 5
        If PressKeyDown Then .y = .y + 5
        If PressKeyLeft Then .x = .x - 5
        If PressKeyRight Then .x = .x + 5
        If .x < 24 Then .x = 24
        If .x > 616 Then .x = 616
        If .y < 24 Then .y = 24
        If .y > 456 Then .y = 456
        If PressKeyZ And Jiki.Frame = 0 Then
            '弾発射
            For i = 0 To 4
                For j = 0 To 49
                    If Jshot(j).Use = False Then
                        With Jshot(j)
                            .Use = True
                            .x = Jiki.x
                            .y = Jiki.y
                            .vx = Cos((250 + i * 10) * 0.017453292517) * 16
                            .vy = Sin((250 + i * 10) * 0.017453292517) * 16
                            .t = 250 + i * 10
                        End With
                        Exit For
                    End If
                Next
            Next
            .Frame = 5  '自機はFrameを弾発射用ウェイトとして使う
        End If
        If PressKeyX = 1 And Bom > 0 Then
            'ボム
            For i = 0 To 255
                Enemy(i).Use = False
            Next i
            frmMain.picBack.ForeColor = vbWhite
            frmMain.picBack.Line (0, 0)-(639, 479), , BF
            Bom = Bom - 1
        End If
        If .Frame > 0 Then .Frame = .Frame - 1
    End With
End Sub

Public Sub DrawJiki()
    '描画
  With Jiki
    Call BitBlt(frmMain.picBack.hDC,.x-24,.y-24,48,48,frmMain.picChar.hDC,48,0,vbSrcAnd)
    Call BitBlt(frmMain.picBack.hDC,.x-24,.y-24,48,48,frmMain.picChar.hDC,0,0,vbSrcPaint)
  End With
End Sub

'=============
'   自機弾
'=============
Private Sub MoveJshot()
    Dim i As Long
    Dim j As Long
    For i = 0 To 49
        If Jshot(i).Use Then
            With Jshot(i)
                .x = .x + .vx
```

★最近のベーマガおもしろいです。ｉアプリやワンダーウィッチ，Java…など，1冊で一気に勉強できる雑誌は他にはない！これからも他にはない記事お願いします。（富山市・**よろぴく18歳**）……**【編**：ありがたいお言葉。ますます燃えてきます。**影**：メラメラメラ。**】**

```
                .y = .y + .vy
                If .y < -16 Then .Use = False
                'あたり判定
                For j = 0 To 255
                    If Enemy(j).Use And Enemy(j).t = 0 Then
                        If Sqr((Enemy(j).x - .x) ^ 2 + (Enemy(j).y - .y) ^ 2) < 32 Then
                            Enemy(j).Shield = Enemy(j).Shield - 1
                            .Use = False
                            Score = Score + 10
                        End If
                    End If
                Next j
            End With
        End If
    Next
End Sub

Private Sub DrawJshot()
    Dim i As Long
    With frmMain.picBack
        .DrawWidth = 4
        .ForeColor = RGB(0, 240, 64)
    End With
    For i = 0 To 49
        If Jshot(i).Use Then
            With Jshot(i)
                frmMain.picBack.Line (.x + Cos(.t * 0.017453292517) * 16, .y + Sin(.t *
0.017453292517) * 16)-(.x - Cos(.t * 0.017453292517) * 16, .y - Sin(.t * 0.017453292517) * 16)
            End With
        End If
    Next
End Sub
'=========
'   敵
'=========
Private Sub CreateEnemy(t As Long, x As Single, y As Single, vx As Single, vy As Single)
    Dim i As Long
    For i = 0 To 255
        If Enemy(i).Use = False Then
            With Enemy(i)
                .Use = True
                .t = t
                .x = x
                .y = y
                .vx = vx
                .vy = vy
                .Shield = 10
                .Frame = 0
            End With
            Exit For
        End If
    Next i
End Sub

Private Sub MoveEnemy()
    Dim i As Long
    Dim j As Long

    For i = 0 To 255
        If Enemy(i).Use Then
            With Enemy(i)
                .x = .x + .vx
                .y = .y + .vy
                Select Case .t
                    Case 0 '敵
                        '攻撃
                        If .Frame Mod (40 - Level) = 0 Then
                            j = atn2(Jiki.x - .x, Jiki.y - .y)
                            Call CreateEnemy(1, .x, .y, Cos(j * 0.017453292517) * 4, Sin(j *
0.017453292517) * 4)
                            If Level > 9 Then
                                'Level10以上は3Way
                                Call CreateEnemy(1, .x, .y, Cos((j - 60) * 0.017453292517) * 4,
Sin((j - 60) * 0.017453292517) * 4)
                                Call CreateEnemy(1, .x, .y, Cos((j + 60) * 0.017453292517) * 4,
Sin((j + 60) * 0.017453292517) * 4)
                            End If
                        End If
                        '破壊
                        If .Shield < 1 Then
                            Call CreateEnemy(2, .x, .y, 0, 0)
                            .Use = False
                            Score = Score + 100
                        End If
                    Case 1
                        'あたり判定
                        If Jiki.Shield = 0 And Sqr((Jiki.x - .x) ^ 2 + (Jiki.y - .y) ^ 2) < 8
Then
                            'GameOver
                            Jiki.Shield = -1
                            Call CreateEnemy(2, Jiki.x, Jiki.y, 0, 0)
                        End If
                    Case 2 '爆発
                        If .Frame = 20 Then .Use = False
                End Select
                If .x < -24 Or .y < -24 Or .x > 664 Or .y > 502 Then .Use = False
                .Frame = .Frame + 1
            End With
        End If
    Next
End Sub

Private Sub DrawEnemy()
    Dim i As Long
    For i = 0 To 255
        If Enemy(i).Use Then
            With Enemy(i)
                Select Case .t
                    Case 0 '敵
                        Call BitBlt(frmMain.picBack.hDC, .x - 24, .y - 24, 48, 48,
frmMain.picChar.hDC, 144, 0, vbSrcAnd)
                        Call BitBlt(frmMain.picBack.hDC, .x - 24, .y - 24, 48, 48,
frmMain.picChar.hDC, 96, 0, vbSrcPaint)
                    Case 1 '敵弾
                        With frmMain.picBack
                            .DrawWidth = 16
                            .ForeColor = RGB(255, 128, 64)
                        End With
                        frmMain.picBack.PSet (.x, .y)
                    Case 2 '爆発
                        frmMain.picBack.DrawWidth = 64
                        frmMain.picBack.Circle (.x, .y), .Frame * 5, RGB(255 - (.Frame * 10),
0, 0)
                End Select
            End With
        End If
    Next i
End Sub

Private Function atn2(x As Double, y As Double) As Double
    Const PI As Double = 3.14159265359

    If x = 0 And y = 0 Then Exit Function    '--これは問題外

    If x = 0 Then
        If y > 0 Then
            atn2 = 90
        ElseIf y < 0 Then
            atn2 = 270
        End If
        Exit Function
    End If

    atn2 = Atn(y / x) * 180 / PI    '--ここではx <> 0
    '-９０度から９０度の戻り値

    If x < 0 Then atn2 = atn2 + 180

    If atn2 < 0 Then atn2 = atn2 + 360
End Function
```

《リスト３》モジュール basSystem.basのプログラム・リスト

```
Attribute VB_Name = "basSystem"
'====================================
'    DESTRUCTION GOD -システム部分-
'====================================
Option Explicit
```


★ベーマガによると，私は3歳からの読者だそうです。そんなことより，卒業写真に入れるはずのシャープX1model10が，気がつくと捨ててしまわれていました。悲しい！やはりベーマガでも入れるべきか。（兵庫県加古川市・BETTY17歳）……【つぐ美：よろしく！】

```
'==========
'   API
'==========
Public Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
Public Declare Function timeGetTime Lib "winmm.dll" () As Long
Public Declare Function timeBeginPeriod Lib "winmm.dll" (ByVal uPeriod As Long) As Long
Public Declare Function timeEndPeriod Lib "winmm.dll" (ByVal uPeriod As Long) As Long
Public Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
'====================
'   グローバル定数
'====================
'シーン番号
Public Const SN_TITLE As Long = 0   'タイトル表示
Public Const SN_GAME As Long = 1    'ゲーム中
'====================
'   グローバル変数
'====================
Public Frame As Long                'フレームカウンタ
'キーを押しているフレーム数
Public PressKeyUp As Long           '[↑]
Public PressKeyDown As Long         '[↓]
Public PressKeyLeft As Long         '[←]
Public PressKeyRight As Long        '[→]
Public PressKeyZ As Long            '[Z]
Public PressKeyX As Long            '[X]
Public PressKeyEsc As Long          '[Esc]
'====================
'   モジュール変数
'====================
'メインループ
Public MainLoopExit As Boolean 'メインループを抜けるか
Public Scene As Long    'シーン番号
'タイトル
Private index As Long   '選択している項目

'====================
'   システム初期化
'====================
Public Sub SystemInit()
    'メインループ
    MainLoopExit = False
    Scene = SN_TITLE
    Frame = 0
    'キー入力
    Call GetAsyncKeyState(vbKeyUp):     PressKeyUp = 0
    Call GetAsyncKeyState(vbKeyDown):   PressKeyDown = 0
    Call GetAsyncKeyState(vbKeyLeft):   PressKeyLeft = 0
    Call GetAsyncKeyState(vbKeyRight):  PressKeyRight = 0
    Call GetAsyncKeyState(vbKeyZ):      PressKeyZ = 0
    Call GetAsyncKeyState(vbKeyX):      PressKeyX = 0
    Call GetAsyncKeyState(vbKeyEscape): PressKeyEsc = 0
End Sub

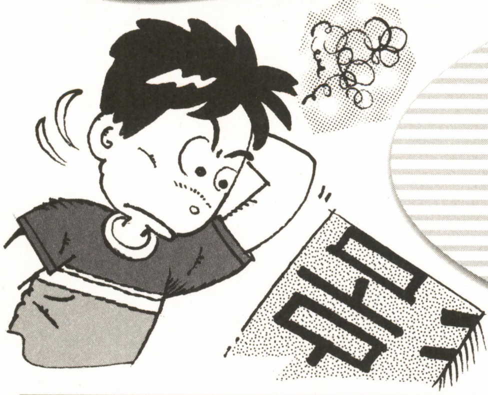
'==================
'   メインループ
'==================
Public Sub MainLoop()
    'ウェイト関係の変数
    Dim FrameStart As Long  'フレーム開始時間
    Dim WaitStart As Long   'ウェイト開始時間
    Dim Wait As Long        'ウェイト時間
    'メインループ
    Do Until (MainLoopExit)
        FrameStart = timeGetTime    '開始時間取得
        frmMain.picBack.Cls         'バックバッファクリア
        '==================
        '   キー入力判定
        '==================
        If GetAsyncKeyState(vbKeyUp) Then PressKeyUp=PressKeyUp+1 Else PressKeyUp=0
        If GetAsyncKeyState(vbKeyDown) Then PressKeyDown=PressKeyDown+1 Else PressKeyDown=0
        If GetAsyncKeyState(vbKeyLeft) Then PressKeyLeft=PressKeyLeft+1 Else PressKeyLeft=0
        If GetAsyncKeyState(vbKeyRight) Then PressKeyRight=PressKeyRight+1 Else PressKeyRight=0
        If GetAsyncKeyState(vbKeyZ) Then PressKeyZ=PressKeyZ+1 Else PressKeyZ=0
        If GetAsyncKeyState(vbKeyX) Then PressKeyX=PressKeyX+1 Else PressKeyX=0
        If GetAsyncKeyState(vbKeyEscape) Then PressKeyEsc=PressKeyEsc+1 Else PressKeyEsc=0
        '==================
        '   シーン別処理
        '==================
        Select Case Scene
            Case SN_TITLE   'タイトル表示
                Call TitleMain
                If PressKeyEsc = 1 Then MainLoopExit = True '[Esc]で終了
            Case Else       'ゲーム中
                Call GameMain
                If PressKeyEsc = 1 Then ChangeScene (SN_TITLE)  '[Esc]でタイトルに戻る
        End Select
        '==============
        '   画面更新
        '==============
        Call BitBlt(frmMain.hDC, 0, 0, 640, 480, frmMain.picBack.hDC, 0, 0, vbSrcCopy)
        frmMain.Refresh
        'ウェイト
        Wait = 25 - timeGetTime + FrameStart    'ウェイト時間計算(40FPS)
        If Wait > 0 Then
            WaitStart = timeGetTime
            Do While (timeGetTime - WaitStart < Wait)
                DoEvents
            Loop
        End If
        Frame = Frame + 1
        DoEvents
    Loop
End Sub

'================
'   シーン変更
'================
Public Sub ChangeScene(SceneNum As Long)
    Scene = SceneNum    'シーン変更
    Frame = -1          '何故-1かというとフレーム終了時に+1されて0になるため
End Sub

'==================
'   タイトル表示
'==================
Public Sub TitleMain()
    If Frame = 0 Then index = 0 '初期化
    '==============
    '   キー入力
    '==============
    If PressKeyZ = 1 Then
        If index Then MainLoopExit = True Else Call ChangeScene(SN_GAME)
    Else
        If PressKeyUp Then index = 0
        If PressKeyDown Then index = 1
    End If
    '==========
    '   描画
    '==========
    With frmMain.picBack
        'タイトル表示
        .FontSize = 48
        .ForeColor = vbRed
        .CurrentX = 80
        .CurrentY = 80
        frmMain.picBack.Print ("DESTRUCTION GOD")
        'メニュー表示
        If index Then
            '[EXIT]を選択
            .ForeColor = RGB(128, 128, 128)
            .CurrentX = 180
            .CurrentY = 196
            frmMain.picBack.Print ("GAMESTART")
            .ForeColor = vbWhite
            .CurrentX = 180
            .CurrentY = 256
            frmMain.picBack.Print ("Exit")
        Else
            '[GAMESTART]を選択
            .ForeColor = vbWhite
            .CurrentX = 180
            .CurrentY = 196
            frmMain.picBack.Print ("GAMESTART")
            .ForeColor = RGB(128, 128, 128)
            .CurrentX = 180
            .CurrentY = 256
            frmMain.picBack.Print ("Exit")
        End If
    End With
End Sub
```

★ルーチンごとの注釈なんて，昔のベーマガのプログラムじゃ，当たり前だったんですけどねぇ。（岩手県和賀郡・**破邪巨星Gバニッシュモンスター29歳**）……**【Dr.D：7月号の68ページのCHECKER FLAGのコメントに関しての発言じゃな。投稿者諸君，よろしく頼むぞ。】**

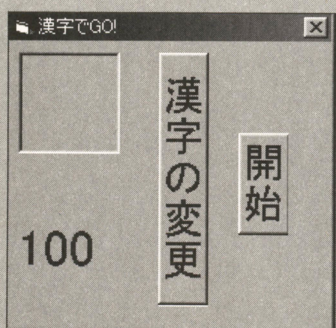
Windows(Visual Basic V6.0)用



この漢字はな～んだ？

漢字(かんじ)で GO(ゴー)!

荒木　伸夫（17歳）



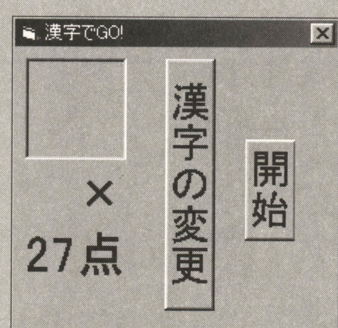
《写真1》制限時間一〇〇秒の漢字を当てるゲーム



《写真2》この拡大部分から漢字を推測



《写真3》各漢字の特徴をよく覚えよう



《写真4》一〇〇秒後に得点が表示される

内容・遊びかた

漢字当てゲームです。漢字の一部（徐々に全体が見えるようになる）が表示されるので，その漢字をクリックしてください。正解すると＋１点，間違えると－１点です。制限時間は100秒です。

プログラム入力時の注意

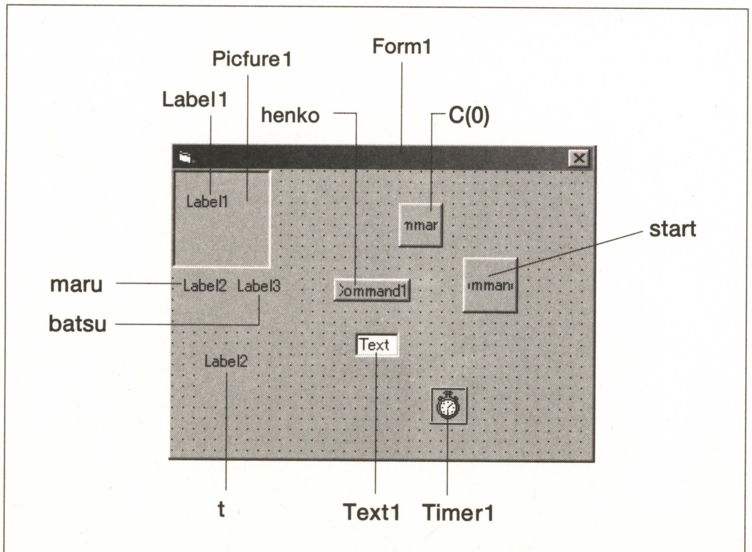
①form1のプロパティのcontrolboxをtrueにする。
②pictureboxを１個，commandbuttonを３個，labelを４個，textboxを１個，timerを１個用意する。
③すべてのcommandbuttonのプロパティの，styleを1-グラフィックスに変更する。
④commandbuttonのオブジェクト名を

《第１表》変数表

a(9)	漢字
i	拡大表示される漢字のindex
x,y	拡大表示される漢字の，どの部分が表示されるかを決定する
tokuten	得点
tim	残り時間
h	漢字を変更する際の，変更中の漢字のindex

《第２表》プログラムの仕組み

(1)ツールのプロパティを定める。
なお，picture1の大きさはlabel1の1/4。
開始がクリックされたら
(2)a( )からランダムで１文字を選ぶ。
(3)label1に(2)で選んだ文字を表示し，ランダムで位置をずらす。
(4)timerを作動させる。
timerで残り時間を計り，徐々に文字の全体が見えるようにする。
(5)クリックされた漢字が正解ならば＋１点，間違いならばー１点。
(6)○，×を表示し，(2)(3)を繰り返す。
(7)残り時間がゼロになったら，ゲーム開始前の状態に戻し，得点を表示する。
漢字の変更（決定）がクリックされたら
(1)text1にa(0)からa(9)を順番に表示。text1の位置も変える。
(2)入力した後，決定をクリックしたら，a( )を変更し，c( )のcaptionも変更する。



フォームの配置図

★実家に初代98と，8インチFDD×2があります。確かDiskBasicのバージョンが2.0とかだった。（東京都町田市・くぼたけ18歳）……
【Dr.D：8インチのフロッピーディスクか，ワシも持っておるぞ。ごそごそ。つぐ美：懐かしい…。いやいや，はじめて見るぅ。】

CHECKER FLAG

Dr.D：このゲームのメインである漢字の拡大処理は，一見難しそうに見えるが，じつはかなり簡単な方法で実現されておる。その方法とは，ピクチャボックスの中に大きいラベルを貼り付けて，そのラベルを動かすことで，漢字の１部が移動しながら拡大表示されている，という仕組みじゃ。以前に編クンが「VBのコントロールの性質を覚えるとリスト簡略化につながる」とコメントしておったが，この作品はまさにその典型じゃな。このコントールの使いかたは見事じゃ！

つぐ美：ただ，ゲームとしてはあらかじめ用意してある漢字の特徴を覚えてしまうとすぐに高得点を出せるようになるので，チョット面白味に欠けますね。ゲームに使用する漢字がランダムで決まるようにするだけでも印象が変わると思いますが。

start，henko，cとし，cのindexを0にする。

⑤labelのオブジェクト名をt，maru，batsu，label1とし，label1をpicture1に貼り付ける。

《第３表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	3 '固定ダイアログ
Height	3570
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	3195
ScaleWidth	4680
ShowInTaskbar	0 'False
StartUpPosition	3 'Windowsの既定値
Top	0
Width	4770
テキストボックスの設定	
Name	Text1
Height	270
Left	2040
TabIndex	8
Text	Text1
Top	1800
Width	495
コマンドボタンの設定	
Name	henko
Caption	Command1
Height	255
Left	1800
Style	1 'グラフィックス
TabIndex	7
Top	1200
Width	855
Name	start
Caption	Command1
Height	615
Left	3240
Style	1 'グラフィックス
TabIndex	6
Top	960
Width	615
Name	C
Caption	Command1
Height	495
Index	0
Left	2520
Style	1 'グラフィックス
TabIndex	2
Top	360
Width	495
ピクチャボックスの設定	
Name	Picture1
Height	1095
Left	0
ScaleHeight	1035
ScaleWidth	1035
TabIndex	0
Top	0
Width	1095
ラベルの設定	
Name	Label1
Caption	Label1
Height	3015
Left	120
TabIndex	1
Top	240
Width	3015
Name	t
Caption	Label2
Height	495
Left	360
TabIndex	5
Top	2040
Width	495
Name	batsu
Caption	Label3
Height	495
Left	720
TabIndex	4
Top	1200
Width	615
Name	maru
Caption	Label2
Height	495
Left	120
TabIndex	3
Top	1200
Width	495
タイマーの設定	
Name	Timer1
Left	2880
Top	2400

『漢字でGO!』のプログラム・リスト

```
Dim a(9) As String '漢字
Dim i As Integer '拡大表示される漢字のindex
Dim x As Integer, y As Integer '拡大表示される漢字の、どの部分が表示されるかを決定する
Dim tokuten As Integer '得点
Dim tim As Integer '残り時間
Dim h As Integer '漢字を変更する際の、変更中の漢字のindex

Private Sub C_Click(Index As Integer)
    If Label1.Caption = C(Index).Caption Then
        tokuten = tokuten + 1
        batsu.Visible = False
        maru.Visible = True
    ElseIf Label1.Caption <> C(Index).Caption Then
        tokuten = tokuten - 1
        maru.Visible = False
        batsu.Visible = True
    End If
    Label1.FontSize = 150
    i = Int(9.9 * Rnd)
    Label1.Caption = a(i)
    Randomize
    x = Int(-2000 * Rnd)
    Randomize
    y = Int(-2000 * Rnd)
    Label1.Move x, y
End Sub

Private Sub Form_Load()
    For j = 1 To 9
        Load C(j)
    Next j
    a(0) = "炎": a(1) = "神": a(2) = "列": a(3) = "忙": a(4) = "仏": a(5) = "禄": a(6) = "持":
a(7) = "穫": a(8) = "狩": a(9) = "品"
    tokuten = 0
    tim = 100
    h = -1
    With Form1
        .Width = 3300
        .Height = 3200
        .Caption = "漢字でGO!"
    End With
    With Label1
        .Width = 3000
        .Height = 3000
        .FontSize = 150
        .Visible = False
    End With
    With Picture1
        .Width = 1000
        .Height = 1000
        .Left = 100
        .Top = 100
    End With
    For j = 0 To 9
        With C(j)
            .Width = 500
            .Height = 500
            .FontSize = 20
            .Caption = a(j)
            .Visible = False
            .Enabled = False
        End With
    Next j
    For j = 0 To 4
        C(j).Move 2000, 100 + 500 * j
    Next j
    For j = 5 To 9
        C(j).Move 2600, 100 + 500 * (j - 5)
    Next j
    With maru
        .Width = 500
```

★OFコーナーに載ったら，CATVのネットサービスが来る，来ない，来る，来ない…（怨）。（広島県東広島市・myatsu田15歳）……
【影：♪来ーる来ない，かーえらない，かーえるぅ。あんの人はいんないのよ，とおい…。**つぐ美：**さて，この歌のタイトルはなんでしょう。】

```
    .Height = 500
    .Left = 100
    .Top = 1200
    .FontSize = 25
    .Caption = ″○″
    .Visible = False
  End With
  With batsu
    .Width = 500
    .Height = 500
    .Left = 600
    .Top = 1200
    .FontSize = 25
    .Caption = ″×″
    .Visible = False
  End With
  With t
    .Width = 1000
    .Height = 500
    .Left = 100
    .Top = 1800
    .FontSize = 25
    .Caption = tim
  End With
  With start
    .Width = 500
    .Height = 1000
    .Left = 2300
    .Top = 900
    .FontSize = 20
    .Caption = ″開始″
  End With
  With henko
    .Width = 500
    .Height = 2500
    .Left = 1500
    .Top = 100
    .FontSize = 20
    .Caption = ″漢字の変更″
  End With
  With Text1
    .Width = 500
    .Height = 500
    .Left = 2300
    .Top = 200
    .FontSize = 20
    .Text = a(0)
    .Visible = False
    .Enabled = False
  End With
  i = Int(9.9 * Rnd)
  Label1.Caption = a(i)
  Randomize
  x = Int(-2000 * Rnd)
  Randomize
  y = Int(-2000 * Rnd)
  Label1.Move x, y
End Sub
Private Sub henko_Click()
  If h <= 8 Then
    h = h + 1
  ElseIf h = 9 Then
    h = 0
  End If
  henko.Caption = ″決定″
  With Text1
    .Visible = True
    .Enabled = True
    .SetFocus
    .Text = a(h)
    .Move C(h).Left, C(h).Top
  End With
End Sub
```

```
Private Sub start_Click()
  tokuten = 0
  Label1.Visible = True
  tim = 100
  t.Caption = tim
  Timer1.Interval = 1000
  With start
    .Visible = False
    .Enabled = False
  End With
  For j = 0 To 9
    With C(j)
      .Visible = True
      .Enabled = True
    End With
  Next j
  maru.Visible = False
  batsu.Visible = False
  With henko
    .Visible = False
    .Enabled = False
  End With
  With Text1
    .Visible = False
    .Enabled = False
  End With
End Sub
Private Sub Text1_LostFocus()
  a(h) = Text1.Text
  C(h).Caption = a(h)
End Sub
Private Sub Timer1_Timer()
  tim = tim - 1
  t.Caption = tim
  If Label1.Left <= -200 Then
    Label1.Left = Label1.Left + 200
  ElseIf Label1.Left >= 200 Then
    Label1.Left = Label1.Left - 200
  ElseIf Label1.Left > -200 And Label1.Left < 200 Then
    Label1.Left = 0
  End If
  If Label1.Top <= -200 Then
    Label1.Top = Label1.Top + 200
  ElseIf Label1.Top >= 200 Then
    Label1.Top = Label1.Top - 200
  ElseIf Label1.Top > -200 And Label1.Top < 200 Then
    Label1.Top = 0
  End If
  If Label1.FontSize >= 60 Then Label1.FontSize = Label1.FontSize - 6
  If tim = 0 Then
    With start
      .Visible = True
      .Enabled = True
    End With
    For j = 0 To 9
      With C(j)
        .Visible = False
        .Enabled = False
      End With
    Next j
    t.Caption = tokuten & ″点″
    Label1.Visible = False
    With henko
      .Visible = True
      .Enabled = True
      .Caption = ″漢字の変更″
    End With
    h = -1
    Timer1.Interval = 0
  End If
End Sub
```


★特選プログラムコーナーのコメントが最近きちんとしてきたので良いです。近頃のベーマガはやまかんさんや，松原さんなど，かつての愛読者に支えられている感じですね。がんばってください。（東京都府中市・MJN27歳）……【編：君たちも筆者になれるんだよ。】

リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0)用



好きな画像をロジック化できる

Logicer
(ロジッカー)

usk（16歳）

内容と使いかた

ビットマップからロジックが作れるという代物です。

まず，ロジックにしたい画像を選択します。一応ビットマップからということになってますが，だいたいの画像(jpegなど)は読み込めます。つぎに，“濃さ”・“粗さ”・“大きさ”・“解答を表示”・“白黒反転”・“詰める”を選択します。“濃さ”は0から765まで指定でき，数が小さいほど白くなります。“粗さ”は1から20を指定でき，数が小さいほど元の画像が鮮明になります。大きい画像でここの数字が細すぎると処理しきれない場合があります。“大きさ”は2から20を指定でき，ロジックにしたときのマスの大きさを決めます。“解答を表示”はチェックすると解答を表示します。濃さなどを調整する場合はチェックをはずさないでください。“白黒反転”は，ロジックにしたとき白と黒を反転します。

簡単な文字をロジックにするときは，チェックするとかなり簡単になってしまいます。“詰める”は1列が完全に白または黒のとき，端をカットします。あと，ロジックを微修正できます。マスの部分をクリックすると白と黒が反転します。

気に入ったロジックができるまでその操作を繰り返してください。

できたら，ビットマップで保存します。保存したファイルを印刷するなり，そのまま解くなり自由です。

微修正以外メニューを開くことなく，すべてキーボードで操作できます。

プログラムの仕組み

まず，読み込んだ画像を“粗さ”で指定された数字を一辺とする正方形で分割します。その正方形の1ピクセルずつRGBを読み込み，そのRGBすべてを足します。マスの中でこれを繰り返し，平均値を出します。これを濃さとし，“濃さ”で指定された数字以下だった場合，黒になります。これを繰り返していきます。“粗さ”と画像の大きさによって余ってしまう部分がありますが，そこは切り取ってしまいます。

つぎに，表示する数字を調べます。一番右下から上に向かって白だったら無視し，黒だったら変数に1を足し，その変数が1以上で白だった場合，または調べているマスが一番上だった場合，記録します。これを上の数字，左の数字を調べます。このとき，表示する数字の数も調べておきます。最後に数字を書き，ロジックを描き，線を引きます。

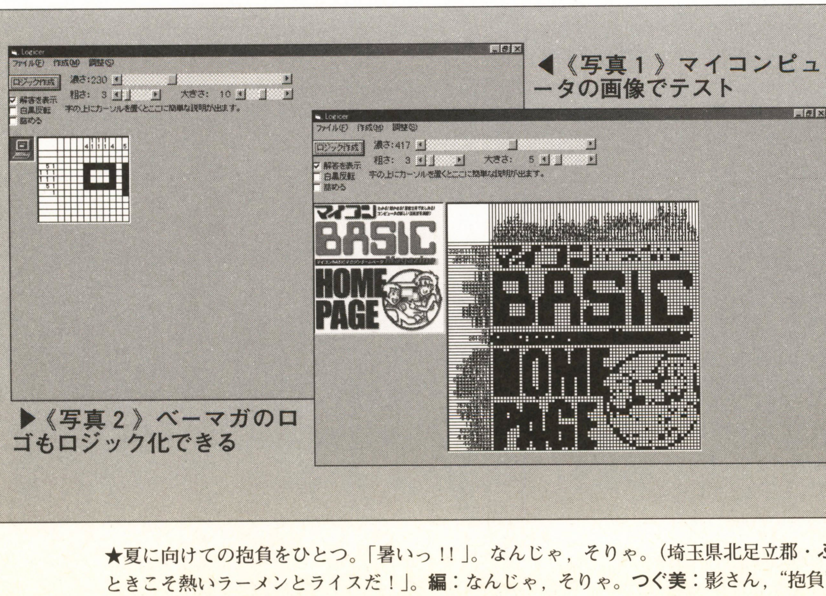
CHECKER FLAG

つぐ美：ロジックをプレイするソフトや，解答を求めるソフトは過去の投稿作品でも紹介してきましたが，画像ファイルからロジックの問題を作成するというのは，新しい形ですね。

Dr.D：プログラム的には，画像処理の内容をしっかりと取り入れないといけないので高度な作品と言える。実用性では，使用者側が何回か調整しないと納得いく解答にならないので，チト扱いが面倒な面もあるが，面白味のあるツールだ。ロジックの好きな読者は1度は使ってみる価値があるゾ。

改造と苦労点

改造したいところは，印刷できるようにすることですね（プリンタがないので実験ができない)。また苦労点といえば，数字を調べるというのに手間をかけさせられました。しかし，しょうもないミスを犯す自分にとって，わりとすんなりいけたほうではないかと思います。



◀《写真1》マイコンピュータの画像でテスト

▶《写真2》ベーマガのロゴもロジック化できる

★夏に向けての抱負をひとつ。「暑いっ!!」。なんじゃ，そりゃ。（埼玉県北足立郡・ぶんちょ26歳）……【影：では私も抱負を。「暑いときこそ熱いラーメンとライスだ！」。編：なんじゃ，そりゃ。つぐ美：影さん，“抱負”という言葉の意味知ってるんですか？もう。】

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0+DirectX7)用



DirectX技術を使った3D作品

Pandora Bottle（パンドラ ボトル）

マーサン（15歳）

ゲーム概要

Direct3D RMを使用した，激遅3Dガンシューティング・ゲームです。怪物が迫ってくるので，マウスの左クリックで射撃，右クリックで弾補充です。規定の得点を獲得すると，エンディングも迎えられます。

ストーリー

1人の男がいた。海の男らしい。男は海岸でビンを見つけコルクを抜いたようだ…。だが“それは，決して開けてはならないビンだった…”。

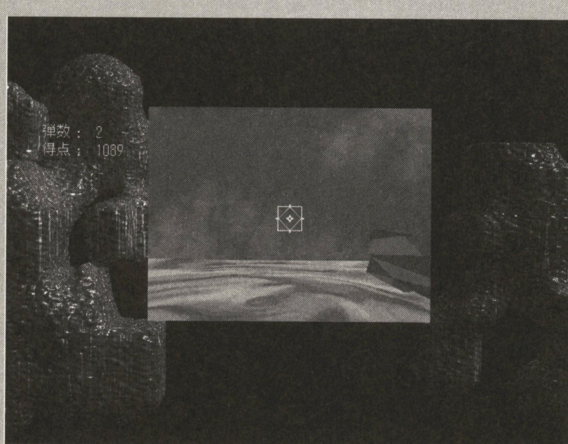
海の男改め，ついてない男は深呼吸し，おもむろに銃を構えた。

遊びかた

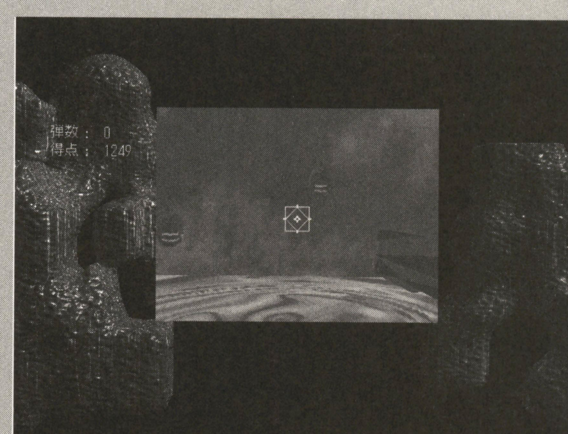
PB.exeをクリックしてください。ゲーム開始です。

あなたは海の男改め，ついていない男として怪物と戦います。

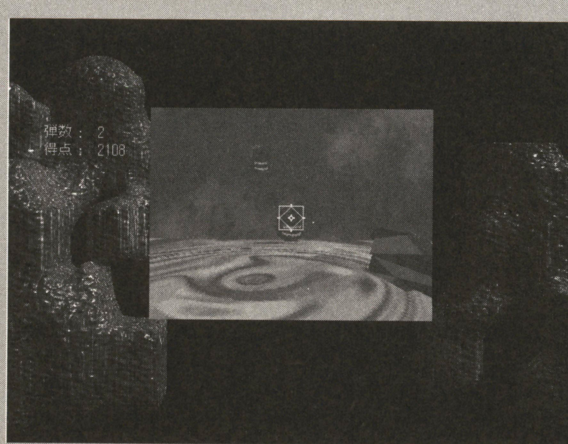
マウスを動かして照準を敵に合わせ，左クリックで攻撃します。クリックした



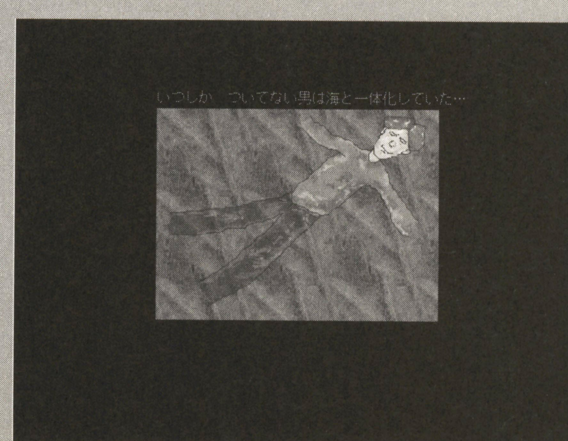
《写真1》3D形式のシューティング



《写真2》不気味な敵キャラが接近してくる



《写真3》照準を合わせて弾を連射だ



《写真4》何回か敵に接近されるとゲーム・オーバー

★夏になり，事故って入院，しょっちゅう見舞い。（長野県松本市・TAC25歳）……【影：“夏になり，Dr.の頭で，蚊がすべる”。おそまつ。Dr.D：座布団，全部持ってケー。編：“夏になり，つぐ美ちゃんの色気で，蚊もコロリ”。つぐ美：座布団，全部編さんに！】

CHECKER FLAG

Dr.D：リスト的には3D処理を使用しており，高度な内容と言えるが，ゲームとしてみると全体的に単調で，シューティングの魅力である爽快感や緊迫感に欠ける印象じゃな。改良するとすれば，やはり実行速度ゃ。ここを改善すれば前述の不満点も解消してくるじゃろう。

つぐ美：しばらく休んで気合いを入れ直したら，またベーマガへ投稿してきてくださいね。それまでに数学をしっかり学んで三角関数と3Dをやるなら4×4行列計算は理解しておきましょう。

編：3Dプログラムは難しいですが，使いこなせると魅力的な作品ができます。自分のペースで徐々に勉強していってはいかがでしょうか？

ままでの連射はできません。弾が足りなくなったら，右クリックで弾を補充してください。

弾は無限に使用できますが，弾を補充するたびに敵の移動スピードが速くなりますので気をつけてください。しかし，敵のスピードが速ければ速いほど，敵を倒したときの得点が増します。敵があるところまで接近すると体力が減り，一時的に動けなくなります。体力がなくなればゲーム・オーバーです。

あるスコア以上で敵にやられたとき，エンディングを迎えることができます。まったく大した物ではないので，失望される方もいるかと思いますが，なんとかお楽しみください。ぜひ頑張ってエンディングを迎えてください。

目指せ20000点!! ちなみに僕のハイスコアは"11342"点です。頑張って僕の記録を抜いてください。

コンフィグ

ゲーム環境を変更したい場合,configフォルダにあるconfig.txtを編集してください。

1行目は，BGMの設定(半角大文字“ON”か“OFF”)。2行目は，ウェイトの設定(半角“0”{ノーウェイト},“1”{少々のウェイト},“2”{かなりのウェイト})。3行目は,エンディング処理(“ON”か“OFF”)となっています。

初期値では，BGMオン，ノーウェイト，エンディングあり，となっています。なお，ウェイトは簡易的なものなので，あまり過度な期待はしないでください。

不備と苦労

努力しましたが解決できなかった不備があります。

・MIDIが鳴らないときがある。

・プレイヤーがダメージを受けたときのポリゴン板が目の前に来ない(←これは，カメラの子フレームとして設定していないから)。

二つ目の不備は，子フレームとしてセットし，カメラといっしょに回すと変な所へ行ってしまうので，子とせず大き目のポリゴンをセットしました。ですので，カメラが端のほうへ行くと画面全体にエフェクトがかかりません。

苦労したことはたくさんあります。序盤だったからよかったものの「レジストリエラー」でWindows初期化し，今までの苦労がパァになったり，3D関係のわからんことが多すぎで大変でした(泣き寝入りしましたし)。

他のいろいろ

汚いプログラムソースで大変恐縮ですが，興味のある方は見てください(DirectInputで，ESCキーの制御しかやってなかったりと，かなりアホなことをやってます)。ポリゴンモデルの作成は『六角大王』と『メタセコイアLE』を使っています。両ソフトの作者に深く感謝いたします。

最後に

このゲームは，参考にし『DirectX7 FOR Visual Basic はじめるゲームプログラミング』に収録されている，サンプルゲームと酷似していました。それでは丸パクリなので，何か独自性を付けようということで色々悩んだ末に，“DirectInput(マウス)”を使おうと思いました。マウスを使うだけでちょっとは違った雰囲気になったかなと思っています。大変になることは必至だったけれど，頑張ってエンディングも入れました。

プログラムで，この上ないバグの量に当惑し，気が滅入ってしまいました。

伊藤隆朗さんという方が僕と同い年のようですね。『レーザーシュートIII』カッコ良かったですよー。大変な技術力があってうらやましいなぁ。もうプログラムには触りたくない気分&視力低下で，投稿もしばらくないと思うので，ベーマガのDirectX 投稿はあなたに託しました。頑張ってねー。

エンディングがかなり簡易的なものなので，あまりに淡白で失望される方もいらっしゃるかと思います。けれども，「そんなことなかったよ」なぁーんて方がいらっしゃったら，苦労した甲斐があった，という気分でございます。

でも，コサイン，サインも知らないようなヘッポコプログラマーが“3D”に手を出すのは無謀だったような気がします。でも完成できて良かったよー。そんじゃ(死語)，さよならー(満面の笑みで)。

★参考にしたもの

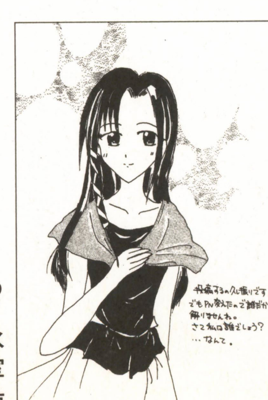
・DirectX7 FOR VisualBasicはじめるゲームプログラミング 藤田伸二著 翔泳社

・ベーマガ(GET&RUNとDirectXプログラミング) 山崎寛，平井昌人



by 神奈月

評…ぐはは、私の季節だ！ぐははは（影）



by 氷室涼

評…♪髪の長い女だって？（影）

★6月号の111ページに久しぶりに登場したなにわさん，自己紹介よろしくお願いします。（神奈川県川崎市・桜井道孝20歳）……【なにわ：えー，マイコンソフト社の開発チームを支えているなにわです。読者のみなさん，これからもよろしく！台湾の地からでした。】

Windows(Excel VBA)用



立方体を使った対戦型パズルゲーム

キューベスト

CUBEST

NAK（28歳）

内容

対戦形式のパズルゲームです。4×4×4のフィールドに交互にキューブを置き，その数を競います。2人で対戦，またはコンピュータと対戦することができます。

ルールの説明

(1)まずタイトル画面で，何人で遊ぶかやコンピュータの強さを選択します。1人で遊ぶ場合，プレイヤーは必ず先手（赤）になります。

(2)キューブは，すでに置かれているキューブの隣に置くことができます。他のキューブと離れた場所や，5段目には置くことができません。

(3)キューブを置くと，そのキューブの隣（前後左右下）にあるキューブの色が，赤なら青，青なら赤に反転します。

(4)このようにして2人交互にキューブを置いていきます。4×4×4のフィールドがすべて埋まった時点で，キューブの多いほうが勝ちとなります。

《第1表》変数表

AKI	思考ルーチン用
BX,BY,BZ	コンピュータが置く位置
CX,CY,CZ	カーソル位置
D(,,)	フィールドの状態
DD(,,)	フィールド一時保存用
DMY	変数一時保存用
E(,)	思考ルーチン用
H(,)	フィールドの高さ
LV	難易度
OLDCX,OLDCY,OLDCZ	移動前のカーソル位置
PL	プレイヤー
SCBLUE,SCRED	スコア
VX,VZ	カーソル移動方向

操作方法

操作はすべて画面上のボタンをクリックすることで行います。

・カーソル移動

↑，↓，←，→ボタンでカーソルを動かします。

・キューブ設置

■ボタンでカーソルのある位置にキューブを置きます。キューブを置けない位置にカーソルがあるときは，ボタンの表示が×になります。

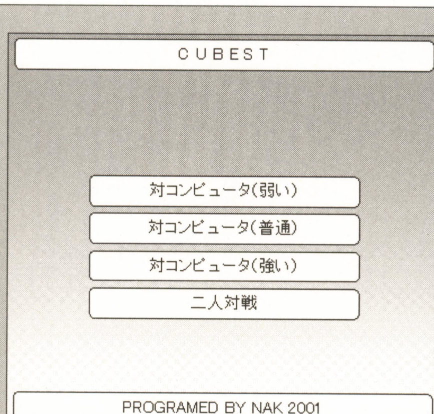
また2人対戦のとき，どちらが置く番なのかを色で判断することができます。

・視点切替

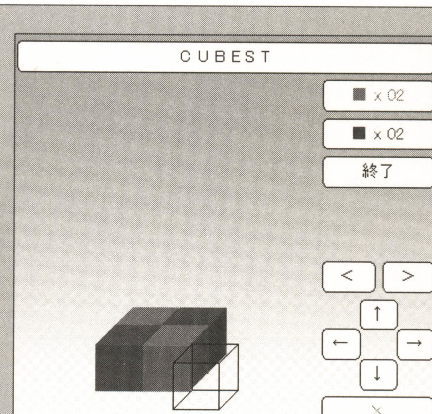
＜，＞ボタンを押すと，フィールド全体が回転します。

・ゲーム終了

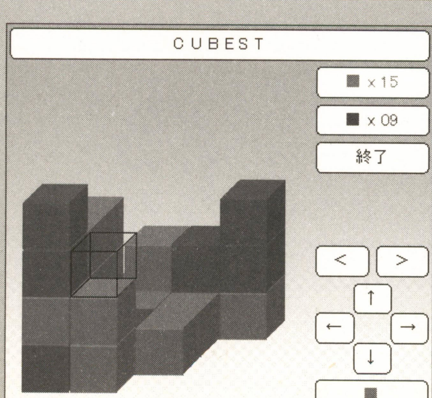
終了ボタンを押すとゲームを中止してタイ



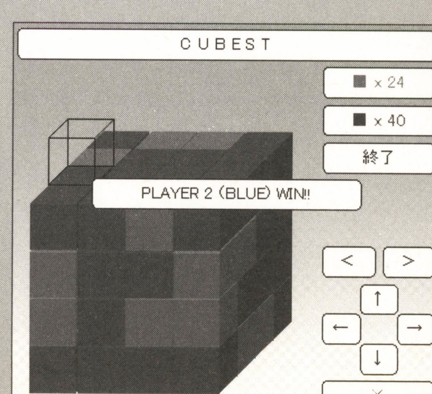
《写真1》COM戦か2人対戦かを選択



《写真2》立方体を利用した対戦パズル



《写真3》対COM戦はかなり白熱するゾ



《写真4》フィールドがすべて埋まると対戦終了

★あのドクペがついに我が町にも上陸。さっそく飲んでみましたが，半分ほどで気分が悪くなりました。恐るべしドクペ。（北海道札幌市・春陽19歳）……【つぐ美：ちょっと時間がたつと，なんとなく飲みたくなるんだってよ。その後どう？影：ビールは「銀河」！】

CHECKER FLAG

編：先月掲載した「４つ並び」のような，Excelのセルを利用した作品とは違う形式のVBA作品ですね。単純な内容ながらも，視点変更ができるグラフィック部分など見栄えのする作品です。

Dr.D：この手のゲーム内容の本質は，３次元配列を利用すれば簡単にできるものじゃが，その内容をわかりやすく表現するのが難しい。やはり，この作品のように疑似３Ｄ形式の表示が望ましいな。

つぐ美：VBA関連でこういったゲーム投稿が増えていくとOfficeのソフトを使った仕事中につい遊んでしまいそうね。

トル画面に戻ります。

入力・実行の手順

(1)Excelを起動します。

(2)メニューからツール→マクロ→Visual Basic Editorと選択して，Visual Basicの画面を表示します。

(3)メニューから挿入→標準モジュールと選択して，Module1を挿入します。

(4)Module1にリストを入力します。

(5)実行する場合，まずVisual BasicからExcelの画面に戻ります。

(6)メニューからツール→マクロ→マクロと選択して，マクロ名の一覧を表示します。

(7)「TITLE」というマクロを選択し，実行ボタンを押すと，タイトル画面が表示されます。

その他

「Excel VBAで無理矢理ゲームを作ろう」をテーマに，使えそうな機能をいろいろと模索した結果，こんなゲームが完成しました。コンピュータの思考ルーチンがうまくできたので気に入っています。作者をも負かしてしまうコンピュータに，ぜひ挑戦してみてください。

☆　☆　☆

```
With ActiveSheet.Shapes.AddShape(5, 15, 15, 280, 20)
    .TextFrame.Characters.Text = "C U B E S T"
    .TextFrame.HorizontalAlignment = xlCenter
End With
With ActiveSheet.Shapes.AddShape(5, 220, 40, 75, 20)
    .Name = "REDSCORE"
    .TextFrame.Characters.Text = "■ x 02"
    .TextFrame.Characters.Font.ColorIndex = 3
    .TextFrame.HorizontalAlignment = xlCenter
End With
With ActiveSheet.Shapes.AddShape(5, 220, 65, 75, 20)
    .Name = "BLUESCORE"
    .TextFrame.Characters.Text = "■ x 02"
    .TextFrame.Characters.Font.ColorIndex = 5
    .TextFrame.HorizontalAlignment = xlCenter
End With
With ActiveSheet.Shapes.AddShape(5, 220, 90, 75, 20)
    .TextFrame.Characters.Text = "終了"
    .TextFrame.HorizontalAlignment = xlCenter
    .OnAction = "TITLE"
End With
```

『CUBEST』のプログラム・リスト

```
Option Explicit '変数の宣言を強制
Public D(7, 7, 7), DD(7, 7, 7) As Integer 'フィールドの状態
Public CX, CY, CZ              As Integer 'カーソル位置
Public LV                      As Integer '難易度
Public PL                      As Integer 'プレイヤー

Sub TITLE() '■タイトル画面
    Call ALLDEL '画面クリア
    With ActiveSheet.Shapes.AddShape(1, 10, 10, 290, 265)
        .Fill.ForeColor.SchemeColor = 49
        .Fill.TwoColorGradient msoGradientHorizontal, 1
    End With
    With ActiveSheet.Shapes.AddShape(5, 15, 15, 280, 20)
        .TextFrame.Characters.Text = "C U B E S T"
        .TextFrame.HorizontalAlignment = xlCenter
    End With
    With ActiveSheet.Shapes.AddShape(5, 65, 105, 180, 20)
        .TextFrame.Characters.Text = "対コンピュータ（弱い）"
        .TextFrame.HorizontalAlignment = xlCenter
        .OnAction = "EASY"
    End With
    With ActiveSheet.Shapes.AddShape(5, 65, 130, 180, 20)
        .TextFrame.Characters.Text = "対コンピュータ（普通）"
        .TextFrame.HorizontalAlignment = xlCenter
        .OnAction = "NORMAL"
    End With
    With ActiveSheet.Shapes.AddShape(5, 65, 155, 180, 20)
        .TextFrame.Characters.Text = "対コンピュータ（強い）"
        .TextFrame.HorizontalAlignment = xlCenter
        .OnAction = "HARD"
    End With
    With ActiveSheet.Shapes.AddShape(5, 65, 180, 180, 20)
        .TextFrame.Characters.Text = "二人対戦"
        .TextFrame.HorizontalAlignment = xlCenter
        .OnAction = "TAISEN"
    End With
    With ActiveSheet.Shapes.AddShape(5, 15, 250, 280, 20)
        .TextFrame.Characters.Text = "PROGRAMED BY NAK 2001"
        .TextFrame.HorizontalAlignment = xlCenter
    End With
    ActiveSheet.Protect 'シート保護
End Sub

Sub EASY() '■対コンピュータ（弱い）
    LV = 1: Call INIT
End Sub

Sub NORMAL() '■対コンピュータ（普通）
    LV = 2: Call INIT
End Sub

Sub HARD() '■対コンピュータ（強い）
    LV = 3: Call INIT
End Sub

Sub TAISEN() '■二人対戦
    LV = 0: Call INIT
End Sub

Sub INIT() '■初期設定
    Dim X, Y, Z As Integer 'ループ用
    Randomize '乱数系列の設定
    Application.ScreenUpdating = False '画面更新オフ
    Call ALLDEL '画面クリア
    With ActiveSheet.Shapes.AddShape(1, 10, 10, 290, 265)
        .Line.Weight = 1.5
        .Fill.ForeColor.SchemeColor = 49
        .Fill.TwoColorGradient msoGradientHorizontal, 1
    End With
```

★ワンダーボーグって，ス○ッチャーが使っていたイン…ハ，ハクション！…？クシャミが…？（愛知県名古屋市・ズオウ・リアキ28歳）……【Dr.D：うーむ，伏せ字やら…が多くって，何がなにやら，わかららん。スッチー？スッチーは好きじゃが。編：Dr.が壊れた！】

```
        With ActiveSheet.Shapes.AddShape(5, 220, 160, 35, 20)
            .TextFrame.Characters.Text = "<"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "TURNLEFT"
        End With
        With ActiveSheet.Shapes.AddShape(5, 260, 160, 35, 20)
            .TextFrame.Characters.Text = ">"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "TURNRIGHT"
        End With
        With ActiveSheet.Shapes.AddShape(5, 245, 185, 25, 20)
            .TextFrame.Characters.Text = "↑"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "MOVEUP"
        End With
        With ActiveSheet.Shapes.AddShape(5, 245, 225, 25, 20)
            .TextFrame.Characters.Text = "↓"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "MOVEDOWN"
        End With
        With ActiveSheet.Shapes.AddShape(5, 220, 205, 25, 20)
            .TextFrame.Characters.Text = "←"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "MOVELEFT"
        End With
        With ActiveSheet.Shapes.AddShape(5, 270, 205, 25, 20)
            .TextFrame.Characters.Text = "→"
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "MOVERIGHT"
        End With
        With ActiveSheet.Shapes.AddShape(5, 220, 250, 75, 20)
            .Name = "PUTBUTTON"
            .TextFrame.Characters.Text = "×"
            .TextFrame.Characters.Font.ColorIndex = 3
            .TextFrame.HorizontalAlignment = xlCenter
            .OnAction = "PUTCUBE"
        End With
        For Y = 1 To 5: For Z = 1 To 4: For X = 1 To 4
            With ActiveSheet.Shapes.AddShape(1, _
            40 + X * 32 - Z * 12, 200 + 12 - Y * 32 + Z * 12, 31, 31) 'キューブの描画
                .ThreeD.SetThreeDFormat msoThreeD1
                .ThreeD.PresetLightingDirection = msoLightingTop
                .Name = "BOX" & X & Y & Z
            End With
        Next: Next: Next
        For Z = 0 To 5: For X = 0 To 5: For Y = 0 To 5
            D(X, Y, Z) = 0 '変数の初期化
        Next: Next: Next
        D(2, 1, 2) = 1: D(3, 1, 3) = 1: D(3, 1, 2) = 2: D(2, 1, 3) = 2
        CX = 4: CY = 1: CZ = 4: PL = 1
        Call ALLPUT '全キューブ表示
    End Sub

    Sub TURNLEFT() '■視点切替 (左)
        Dim X, Y, Z As Integer 'ループ用
        Dim DMY     As Integer '変数一時保存用
        For X = 1 To 4: For Y = 1 To 4: For Z = 1 To 4
            DD(X, Y, Z) = D(X, Y, Z) 'フィールドの状態を一時保存
        Next: Next: Next
        For X = 1 To 4: For Y = 1 To 4: For Z = 1 To 4
            D(X, Y, Z) = DD(5 - Z, Y, X) 'フィールドの回転
        Next: Next: Next
        DMY = CX: CX = CZ: CZ = 5 - DMY 'カーソル位置の回転
        Call ALLPUT '全キューブ表示
    End Sub

    Sub TURNRIGHT() '■視点切替 (右)
        Dim X, Y, Z As Integer 'ループ用
        Dim DMY     As Integer '変数一時保存用
        For X = 1 To 4: For Y = 1 To 4: For Z = 1 To 4
            DD(X, Y, Z) = D(X, Y, Z) 'フィールドの状態を一時保存
        Next: Next: Next
        For X = 1 To 4: For Y = 1 To 4: For Z = 1 To 4
            D(X, Y, Z) = DD(Z, Y, 5 - X) 'フィールドの回転
        Next: Next: Next
        DMY = CX: CX = 5 - CZ: CZ = DMY 'カーソル位置の回転
        Call ALLPUT '全キューブ表示
    End Sub

    Sub MOVEUP() '■カーソル移動 (奥)
        Call MOVECURSOL(0, -1)
    End Sub

    Sub MOVEDOWN() '■カーソル移動 (手前)
        Call MOVECURSOL(0, 1)
    End Sub

    Sub MOVELEFT() '■カーソル移動 (左)
        Call MOVECURSOL(-1, 0)
    End Sub

    Sub MOVERIGHT() '■カーソル移動 (右)
        Call MOVECURSOL(1, 0)
    End Sub

    Sub MOVECURSOL(ByVal VX As Integer, ByVal VZ As Integer) '■カーソル移動
        Dim OLDCX, OLDCY, OLDCZ As Integer '移動前の位置
        '画面外には移動不可
        If CX + VX < 1 Or CX + VX > 4 Or CZ + VZ < 1 Or CZ + VZ > 4 Then Exit Sub
        OLDCX = CX: OLDCY = CY: OLDCZ = CZ
        CX = CX + VX: CZ = CZ + VZ: CY = 5
        Do Until D(CX, CY - 1, CZ) <> 0 Or CY = 1 '縦の位置を決める
            CY = CY - 1
        Loop
        Call CHRPUT(OLDCX, OLDCY, OLDCZ)
        Call CHRPUT(CX, CY, CZ)
        Call PUTCHECK 'キューブ設置ボタンの設定
    End Sub

    Sub PUTCUBE() '■キューブを置く
        '置けない場合中止
        If ActiveSheet.Shapes("PUTBUTTON").TextFrame.Characters.Text = "×" Then Exit Sub
        Application.ScreenUpdating = False '画面更新オフ
        Call REVCUBE(CX + 1, CY, CZ) '隣のキューブの色を反転
        Call REVCUBE(CX - 1, CY, CZ)
        Call REVCUBE(CX, CY, CZ + 1)
        Call REVCUBE(CX, CY, CZ - 1)
        Call REVCUBE(CX, CY - 1, CZ)
        D(CX, CY, CZ) = PL
        CY = CY + 1 'カーソルを一段上に移動
        PL = 3 - PL '相手の番
        Call CHRPUT(CX, CY, CZ)
        Call CHRPUT(CX, CY - 1, CZ)
        Call PUTCHECK 'キューブ設置ボタンの設定
        Call SCORECOUNT 'スコア表示・ゲームオーバー
        Application.ScreenUpdating = True '画面更新オン
        If LV > 0 Then Call COMTHINK 'コンピュータ思考ルーチン
    End Sub

    '■キューブの色を反転
    Sub REVCUBE(ByVal X As Integer, ByVal Y As Integer, ByVal Z As Integer)
        If D(X, Y, Z) > 0 Then
            D(X, Y, Z) = 3 - D(X, Y, Z)
            Call CHRPUT(X, Y, Z)
        End If
    End Sub

    Sub PUTCHECK() '■キューブ設置ボタンの設定
        ActiveSheet.Unprotect 'シート保護解除
        With ActiveSheet.Shapes("PUTBUTTON")
```


★学校訪問記は，つまらないというより，羨ましい。パソコン部に入って，将棋のプログラムを組み，あーでもないこーでもないと，みんなで悩みたかった。ゲームばっかしてるPC部の弟，もったいない！（愛知県名古屋市・わいち301　27歳）……【影：はいっ。】

```
        If (D(CX + 1, CY, CZ) > 0 Or D(CX - 1, CY, CZ) > 0 Or D(CX, CY, CZ + 1) > 0 Or _
            D(CX, CY, CZ - 1) > 0 Or D(CX, CY - 1, CZ) > 0) And CY < 5 Then
             .TextFrame.Characters.Text = "■" '置ける
        Else
             .TextFrame.Characters.Text = "×" '置けない
        End If
        If PL = 1 Then
            .TextFrame.Characters.Font.ColorIndex = 3 '赤の番
        Else
            .TextFrame.Characters.Font.ColorIndex = 5 '青の番
        End If
    End With
    ActiveSheet.Protect 'シート保護
End Sub

Sub COMTHINK() '■コンピュータ思考ルーチン
    Dim E(7, 7)    As Integer 'フィールド評価用
    Dim H(7, 7)    As Integer 'フィールドの高さ
    Dim BX, BY, BZ As Integer '置く場所
    Dim AKI        As Integer 'フィールド評価用
    Dim X, Y, Z    As Integer 'ループ用
    Dim R          As Integer '乱数
    R = Int(Rnd * 10) + 1
    For X = 1 To 4: For Z = 1 To 4
        Y = 5
        Do Until D(X, Y - 1, Z) <> 0 Or Y = 1 'フィールドの高さを調べる
            Y = Y - 1
        Loop
        H(X, Z) = Y
        If (D(X + 1, Y, Z) > 0 Or D(X - 1, Y, Z) > 0 Or D(X, Y, Z + 1) > 0 Or _
            D(X, Y, Z - 1) > 0 Or D(X, Y - 1, Z) > 0) And Y < 5 Then
            E(X, Z) = 1 'キューブ置ける
            If R < LV * 3 Then '有利な場所か調べる
                AKI = 0
                If D(X + 1, Y, Z) = 0 And X < 4 Then AKI = AKI + 1
                If D(X - 1, Y, Z) = 0 And X > 1 Then AKI = AKI + 1
                If D(X, Y, Z + 1) = 0 And Z < 4 Then AKI = AKI + 1
                If D(X, Y, Z - 1) = 0 And Z > 1 Then AKI = AKI + 1
                If Y < 4 Then AKI = AKI + 1
                If AKI Mod 2 = 0 Then E(X, Z) = 2
            End If
        End If
    Next: Next
    BX = 0: BZ = 0
    For X = 1 To 4: For Z = 1 To 4
        '置く位置を決める
        If (E(BX, BZ) < E(X, Z)) Or (E(BX, BZ) = E(X, Z) And Int(Rnd * 2) > 0) Then
            BX = X: BZ = Z: BY = H(X, Z)
        End If
    Next: Next
    Application.ScreenUpdating = False '画面更新オフ
    Call REVCUBE(BX + 1, BY, BZ) '隣のキューブの色を反転
    Call REVCUBE(BX - 1, BY, BZ)
    Call REVCUBE(BX, BY, BZ + 1)
    Call REVCUBE(BX, BY, BZ - 1)
    Call REVCUBE(BX, BY - 1, BZ)
    D(BX, BY, BZ) = PL
    If CX = BX And CY = BY And CZ = BZ Then 'カーソルがあった場合、一段上へ
        CY = CY + 1
        Call CHRPUT(CX, CY - 1, CZ)
        Call CHRPUT(CX, CY, CZ)
    End If
    PL = 3 - PL
    Call CHRPUT(BX, BY, BZ)
    Call PUTCHECK 'キューブ設置ボタンの設定
    Call SCORECOUNT 'スコア表示・ゲームオーバー
    Application.ScreenUpdating = True '画面更新オン
End Sub

Sub ALLPUT() '■全キューブ表示
```

```
    Dim X, Y, Z As Integer 'ループ用
    Application.ScreenUpdating = False '画面更新オフ
    ActiveSheet.Unprotect 'シート保護解除
    For Y = 1 To 5: For Z = 1 To 4: For X = 1 To 4
        Call CHRPUT(X, Y, Z) 'キューブ表示
    Next: Next: Next
    ActiveSheet.Protect 'シート保護
    Application.ScreenUpdating = True '画面更新オン
End Sub

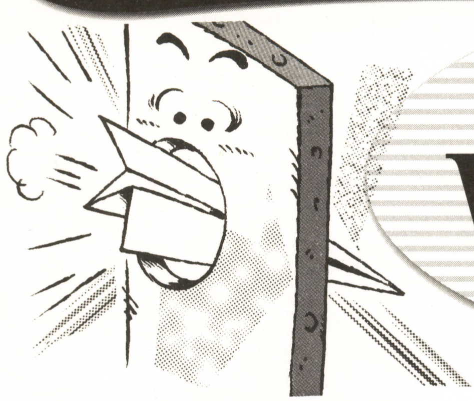
Sub CHRPUT(ByVal X As Integer, ByVal Y As Integer, ByVal Z As Integer) '■キューブ表示
    ActiveSheet.Unprotect 'シート保護解除
    With ActiveSheet.Shapes("BOX" & X & Y & Z)
        If D(X, Y, Z) = 0 Then 'キューブなし
            .Line.Visible = msoTrue
            .ThreeD.Visible = msoFalse
            .Fill.Visible = msoFalse
            .Line.Visible = msoFalse
        Else 'キューブあり
            .Line.Visible = msoTrue
            .Fill.Visible = msoTrue
            .ThreeD.Visible = msoTrue
            .ThreeD.SetThreeDFormat msoThreeD1
            .ThreeD.PresetLightingDirection = msoLightingTop
            Select Case D(X, Y, Z)
                Case 1: .Fill.ForeColor.RGB = RGB(255, 96, 96)
                Case 2: .Fill.ForeColor.RGB = RGB(0, 96, 255)
            End Select
        End If
        If X = CX And Y = CY And Z = CZ Then 'カーソルの表示
            .Line.Visible = msoTrue
            .ThreeD.Visible = msoTrue
            .Fill.Visible = msoTrue
            .ThreeD.SetThreeDFormat msoThreeD4
        End If
    End With
    ActiveSheet.Protect 'シート保護
End Sub

Sub SCORECOUNT() '■スコア表示・ゲームオーバー
    Dim SCRED, SCBLUE As Integer 'スコア
    Dim X, Y, Z       As Integer 'ループ用
    ActiveSheet.Unprotect 'シート保護解除
    For Y = 1 To 4: For Z = 1 To 4: For X = 1 To 4
        Select Case D(X, Y, Z)
            Case 1: SCRED = SCRED + 1 'キューブを数える
            Case 2: SCBLUE = SCBLUE + 1
        End Select
    Next: Next: Next
    'スコア表示
    ActiveSheet.Shapes("REDSCORE").TextFrame.Characters.Text="■ x " & Format(SCRED,"00")
    ActiveSheet.Shapes("BLUESCORE").TextFrame.Characters.Text="■ x " & Format(SCBLUE,"00")
    If SCRED + SCBLUE = 64 Then 'ゲームオーバー
        With ActiveSheet.Shapes.AddShape(5, 65, 115, 180, 20)
            Select Case SCRED - SCBLUE
                Case Is > 0: .TextFrame.Characters.Text = "PLAYER 1 (RED) WIN!!"
                Case Is < 0: .TextFrame.Characters.Text = "PLAYER 2 (BLUE) WIN!!"
                Case 0: .TextFrame.Characters.Text = "DRAW GAME"
            End Select
            .TextFrame.HorizontalAlignment = xlCenter
        End With
    End If
    ActiveSheet.Protect 'シート保護
End Sub

Sub ALLDEL() '■画面クリア
    ActiveSheet.Unprotect 'シート保護解除
    ActiveSheet.Shapes.SelectAll
    Selection.Delete
End Sub
```


★力がほしいか？（茨城県古河市・**年齢不相応41歳**）……【**影：**ははーっ，神様，私に力をお与えください。**神（編集長）：**力を与えてほしくば，本日は1人で残業し，残りの原稿を処理しておくように。**影：**ははーっ。**神：**他の者はビアガーデンに集合！よいな。】

Windows(Visual C++ V6.0)用



DirectXで過去の作品をリメイク

移植版

Walls in the Box

ウォールズ　イン　ザ　ボックス

夏　夕樹（19歳）

内容

95年，電波新聞社刊の「PC9801/9821プログラム大全集Ⅱ」にある佐々木淳哉様の3Ｄワンキー・ゲームのWindows移植版。壁の穴をくぐってゆくゲームです。

遊びかた

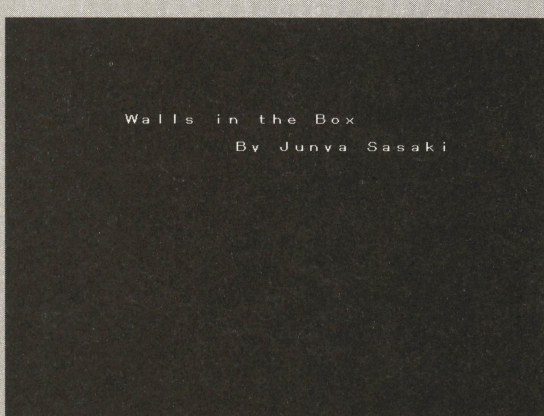
Ｚキーを押すと画面中央の自機が上昇し，離すと下降します。壁がどんどん迫ってくるので，壁の穴をうまくすり抜けてください。壁にぶつかったり，下の床に落ちたりするとミスになります。

進むにつれて画面左上のDISTANCEのメーターが増えていき，メーターがいっぱいになるとステージ・クリアとなります。全５面です（佐々木淳哉様の説明文をそのまま引用させていただきました）。

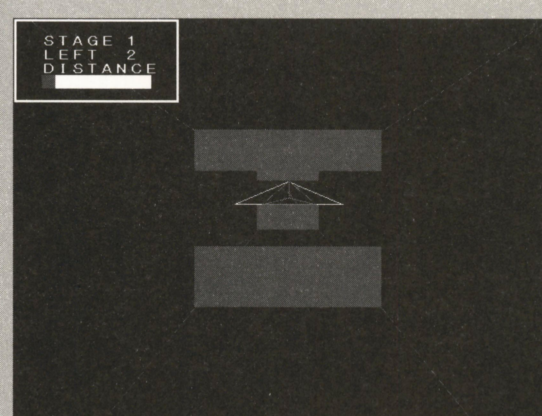
プログラムの仕組み

プログラムもほぼ完全移植(コピー)なので私が言うことではありませんが，3D描画の際に必要な計算を軽減するために，奥行きの距離に対する描画倍率をテーブル化しています。なお移植に際してフレームレートを３倍（約60フレーム）に増やしました。今どきのパソコンならへでもないと思います。

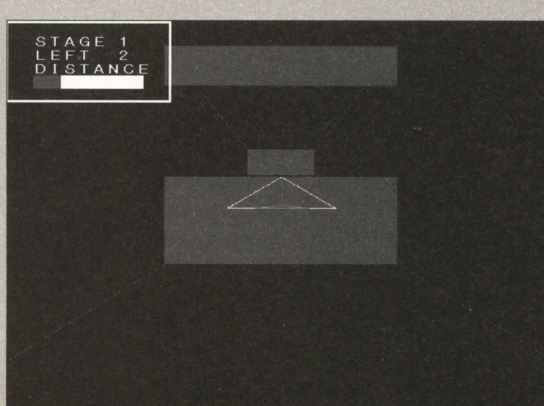
あと，ライブラリ側にBASICでいうところのPAINT機能がないので自機はワイヤーフレーム表示となっています。かなり目劣りしますがお許しください。



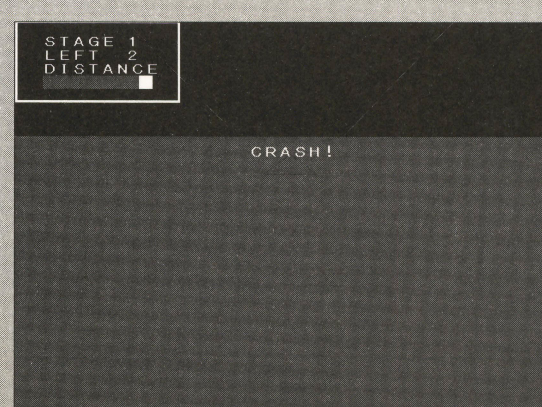
《写真1》懐かしの名作をDirectXを使って再現



《写真2》ワンキーで遊べる、定番の3Dゲーム



《写真3》なめらかな動きが特徴の一つといえる



《写真4》このようにクラッシュしないように

★編集部のみなさん，がんばってください。それから，仕事で「目」を悪くしないように注意してください。（東京都練馬区・畑の木28歳）……【一同：ありがとう，がんばります。読者の励ましが大きな力となります。影：では，目を休ませるために，ちょっと昼寝を。】

さらに元々が，N88日本語BASICのプログラムなので，リストを短くするべくC言語らしからぬ，ものすごいプログラムになってしまいました。N88日本語BASICって単位面積当たりのプログラム情報量がものすごく多いんですね。

プログラムの入力方法・注意点

このプログラムは，私の自作ライブラリであるDXライブラリを使用して作られているので，実行にはDXライブラリを用意する必要があります。DXライブラリは，つぎのURLからダウンロードできます。http://homepage2.nifty.com/natupaji/DxLib/

本格的なゲームも作れるように設計されている2D専用ライブラリです。

DXライブラリを使ったソフトの具体的な作りかたは上記のホームページに載っていますので参照してください。

改造のアドバイス

改造するべきところは少ないですが，壁や自機にテクスチャーを貼る，移動スピードを上げる，音楽や音を付ける，といったところでしょうか？

苦労したところ

やはりオリジナルプログラムの理解です。プログラムを短くするため，処理を軽くするためのからくり的な記述が多かったので少し大変でした。

最後に

96年頃からベーマガを読み始め，出そう出そうと思いつつ結局5年という歳月が過ぎてしまいました。やっと投稿できました。

このプログラムで使用しているDXライブラリは，以前のN88日本語BASICのような使い勝手ができるようにも設計されています。C言語でゲームプログラムをはじめる方は，ぜひ使ってみてください（宣伝）。

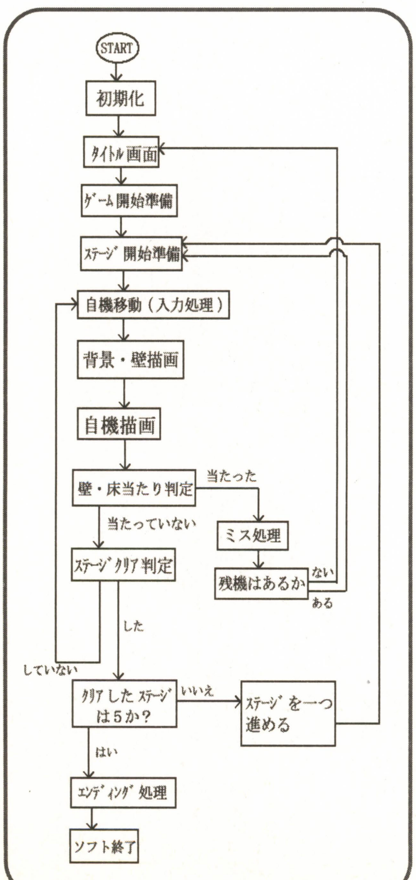
CHECKER FLAG

Dr.D： 過去のPC98などの作品をWindowsへ移植するのは，じつは，けっこう難しかったりする。ライブラリ化した機能を使って，ラインや文字描画命令などを実現しているのが，Windowsへの移植につながっておるな。ただ，DirectX作品の場合は，終了処理を忘れずに備えてほしいものじゃ。

編： DXライブラリは，DirectXの初期化やグラフィックを利用しやすくするために作られたライブラリのようですね。うまく作られているので，とくに初期化などでDirectXプログラムに挫折した読者は，このライブラリを使ってもう一度チャレンジしてみてはどうでしょう？

つぐ美： 最近では，DirectX SDK側でもDirectX3DXという初期化をコンパクトにまとめたライブラリが使えるようになっていたりします。

影： 説明文には書かれていないけど，ジョイスティックでも遊べるゾ。また，速度が遅いと感じる場合は，リストにある“WaitTimer”の値を小さくしてみよう。



フローチャート

『Walls in the Box』のプログラム・リスト

```
//------------------------------------------------
//      Walls in the Box 移植版        by 夏夕樹
//------------------------------------------------

// インクルード部------------
#include <windows.h>
#include <math.h>
#include <stdlib.h>
#include <mbstring.h>
#include "DxLib.h"

// マクロ定義部-------------

// データ型定義-------------

// 自機データ
typedef struct tugPLAEYRDATA
{
    double  A ;                                      // 自機の角度
    int     x1[6],y1[6],x2[6],y2[6];                 // 自機の頂点座標
    double  y ;                                      // 自機の高さ
    double  z1 ;                                     // 自機描画時の高さ計算に使われるデータ
} PLAYERDATA , *LPPLAYERDATA ;

// ソフト全体で使うデータ型
typedef struct tugSYSDATA
{
    int             Cr[16] ;                         // 色データ
    double  v[ 192 ];                                // 距離による描画倍率のデータ
    int             St ;                             // ステージ
    int             Ct ;                             // カウンタ
    int             Di ;                             // 進んだ距離
    int             Lf ;                             // 残機

    int             Wi ;                             // 壁の穴の幅
    int             Pe ;                             // 壁の間隔
    int             H1 , H2 ;                        // 壁の穴の高さ
    int             Df , Fl ;                        // フラグ
    int             C1, C2 ;                         // 壁の位置
} SYSDATA , *LPSYSDATA ;
```


★購入したベーマガの裏表紙に傷がっ！影さん，ついに本の破壊にまで手を出しましたか。（群馬県前橋市・トランス18歳）……【影：どうせ，読みつぶすんでしょ。気にしない気にしない。保管用としてもう1冊買うことをおすすめいたします。どう，解決したでしょ。】

```
// データ宣言部------------------------

// 色取得用データ
int Pal[ 16 ][ 3 ] = {
          {0,0,0},{0,0,255},{255,0,0},{255,0,255},{0,255,0},{0,255,255},
          {255,255,0},{255,255,255},{119,119,119},{0,0,170},{170,0,0},
          {170,0,170},{0,170,0},{0,170,170},{170,170,0},{170,170,170} } ;

// 自機描画用データ
int PlayerData[] = { -63,-63,63,-63,0,63,-31,-63,31,-63,0,-48,31 } ;

SYSDATA         Sys ;           // ソフト全体で使うデータ
PLAYERDATA      PlData ;        // プレイヤーのデータ

// 関数宣言部--------------------------

void GameInit( void ) ;         // ゲーム初期化
void TitleFunc( void ) ;        // タイトル画面
void StageInit( void ) ;        // ゲームスタートの準備を行う関数1
void StartInit( void ) ;        // ゲームスタートの準備を行う関数2
void GameMain( void ) ;         // ゲームメイン
void Crash( void ) ;            // クラッシュ
void StageClear( void ) ;       // ステージクリア
void PlayerDraw( int Co,int P1,int P2,int P3 ); // 自機の描画
void WallDraw( int P1 , int P2 ) ;              // 壁描画
void WallDraw2( int P1 , int P2 ) ;             // 壁描画2
void StateDraw( void ) ;                        // ステータス表示
void StrConv( int ConvData , char *DestStr ) ;  // 数値を2バイト文字に変換
// プログラム部------------------------

void GameInit( void )           // ゲーム初期化
{
        int i , *Data ;

        ClsDrawScreen() ; ScreenFlip() ; ClsDrawScreen() ;  // 画面初期化

        // 色データセット
        for( i = 0 ; i < 16 ; i ++ )
                Sys.Cr[i] = GetColor( Pal[i][0], Pal[i][1], Pal[i][2] ) ;

        // 壁描画用データのセット
        for( i = 0 ; i < 192 ; i ++ )
                Sys.v[ i ] = 16.0 / ( 192 + 8 - i ) ;

        // 自機描画用データセット
        Data = PlayerData ;
        for( i = 0 ; i < 6 ; i ++ )
        {
                PlData.x1[i] = *Data ; Data ++ ;
                PlData.y1[i] = *Data ; Data ++ ;
                PlData.x2[i] = PlData.x1[i] + 320 ;
        }
        PlData.z1 = *Data ;
}

void TitleFunc( void )  // タイトル画面
{
        SetDrawScreen( DX_SCREEN_BACK ) ;               // 描画先を裏画面にセット
        ClsDrawScreen() ; ScreenFlip() ; ClsDrawScreen() ;      // 画面初期化

        // タイトル文字描画
        DrawString( 9*16 , 7*16 , "W a l l s   i n   t h e   B o x" ,Sys.Cr[7] ) ;
        DrawString( 17*16, 9*16 , "B y   J u n y a   S a s a k i" ,Sys.Cr[7] ) ;
        ScreenFlip() ;

        WaitKey() ; // キー待ち
}

// ゲームスタートの準備を行う関数1
void StageInit( void )
{
        PlData.A=-.5; PlData.y = 0 ;
        Sys.Wi=128-Sys.St*20; Sys.Pe=192-Sys.St*24; Sys.H1=0; Sys.H2=0; Sys.Df=0;
}

// ゲームスタートの準備を行う関数2
void StartInit( void )
{
        Sys.St=0 ; Sys.Lf=2 ; Sys.Ct=0 ; Sys.Di=-1 ;
}

void GameMain( void )           // ゲームメイン
{
        int K , Flag , i ;

        // 各数値初期化
        StartInit() ;
        StageInit() ;

        // メインループ
        while( !ProcessMessage() )
        {
                // キー入力処理
                K = GetJoypadInputState( DX_INPUT_KEY_PAD1 ) ;
                if( K & PAD_INPUT_A ) PlData.A -= ( PlData.A > -1 ) * .033333333333 ;
                else  PlData.A += ( PlData.A <  1 ) * .033333333333 ;
                // 高さ変更、補正
                PlData.y -= sin( PlData.A ) * 10.66666 ;
                if( PlData.y>  300){ PlData.A*= -1; PlData.y-=sin(PlData.A)*10.66666;}
                if( PlData.y< -300){
                        PlData.A*= -1; PlData.y-=sin(PlData.A)*10.66666; Sys.Df=1;}

                // 背景描画
                DrawLine( 319,200,0  ,(int)PlData.y+499,Sys.Cr[2]) ;
                DrawLine( 320,200,639,(int)PlData.y+499,Sys.Cr[2]) ;
                DrawLine( 320,199,639,(int)PlData.y-100,Sys.Cr[2]) ;
                DrawLine( 319,199,0  ,(int)PlData.y-100,Sys.Cr[2]) ;

                // 背景処理・描画
                Sys.C1 = Sys.Ct % Sys.Pe ; Sys.C2 = ( Sys.Ct + Sys.Pe / 2 ) % Sys.Pe ;
                if( Sys.C1 == 0 ){Sys.H1 = GetRand(19)*16-160 ; Sys.Di ++ ; Sys.Fl=1; }
                if( Sys.C2 == 0 ){Sys.H2 = GetRand(19)*16-160 ; Sys.Fl =0 ; }
                WallDraw( Sys.C1 + 192 - Sys.Pe , Sys.H1 ) ;
                if(Flag=(Sys.Di==(Sys.St+1)*8-1)) WallDraw2(Sys.C1+192-Sys.Pe,Sys.H1) ;
                if( !Flag || Sys.Fl ) WallDraw( Sys.C2 + 192 - Sys.Pe , Sys.H2 ) ;

                // 自機描画
                for( i=0 ; i<5 ; i++ )
                        PlData.y2[i]=(int)(PlData.y1[i]*sin(PlData.A)/2)+200 ;
        PlData.y2[5]=(int)(PlData.y1[5]*sin(PlData.A)-PlData.z1*cos(PlData.A)/2)+200;
                if( PlData.A <= 0 )
                {
                  PlayerDraw( 7,0,1,2 ) ; PlayerDraw( 8,2,3,5 ) ; PlayerDraw( 6,3,4,5 ) ;
                  DrawLine(PlData.x2[2],PlData.y2[2],PlData.x2[4],PlData.y2[4],Sys.Cr[8]);
                }
                else
                {
                        PlayerDraw( 8,0,1,2 ) ; PlayerDraw( 6,3,4,5 ) ;
                }

                // ステータス描画
                StateDraw() ;
```


★プログラミングして思った。1（いち）とl（エル）の区別がわからない。この区別がはっきりわかるようにして。（埼玉県大里郡・Mass17歳）……【**編**：フォントの問題ですね。編集部内でもそのあたりをなんとかしようと試みています。お待ちください。】

```
		// 表示，諸条件の判定
		Sys.Cr[6]=GetColor( GetRand(255),GetRand(255),GetRand(255) ) ;
		if( Sys.Di >= (Sys.St+1)*8 ) StageClear() ;
		ScreenFlip() ; ClsDrawScreen() ;
		Sys.Ct ++ ;
		if( ( Sys.C1==Sys.Pe-1 && abs(Sys.H1+(int)PlData.y)>Sys.Wi ) ||
			(Sys.C2==Sys.Pe-1 && abs(Sys.H2+(int)PlData.y)>Sys.Wi) || Sys.Df)
			Crash();

		WaitTimer( 17 ) ;
	}
}

void Crash( void )		// クラッシュ
{
	SetDrawScreen( DX_SCREEN_FRONT ) ;
	DrawString( 17*16,9*16,"ＣＲＡＳＨ！",Sys.Cr[7] );
	Sys.Lf-- ; WaitKey() ;
	if( Sys.Lf >= 0 )
	{
		Sys.Ct= Sys.Ct/Sys.Pe*Sys.Pe; Sys.Di-- ;
	}
	else
	{
		ClsDrawScreen() ;
		DrawString( 16*16,15*16,"ＧＡＭＥ　ＯＶＥＲ",Sys.Cr[7] );
		WaitKey() ;	TitleFunc() ; StartInit() ;
	}
	StageInit() ;
	SetDrawScreen( DX_SCREEN_BACK ) ;
}
void StageClear( void )		// ステージクリア
{
	SetDrawScreen( DX_SCREEN_FRONT ) ;
	DrawString( 14*16,9*16,"ＳＴＡＧＥ　ＣＬＥＡＲ！！" , Sys.Cr[7] ) ;
	WaitKey() ;
	ClsDrawScreen() ;
	Sys.St ++ ;Sys.Di=-1;Sys.Ct=0;
	if( Sys.St == 5 )
	{
		DrawString(12*16, 9*16,"ＡＬＬ　ＳＴＡＧＥ　ＣＬＥＡＲ！！",Sys.Cr[7]) ;
		DrawString(12*16,11*16,"ＣＯＮＧＲＡＴＵＬＡＴＩＯＮＳ！！",Sys.Cr[7]) ;
		WaitKey() ; DxLib_End() ;
		exit(0);
	}
	StageInit() ;
	SetDrawScreen( DX_SCREEN_BACK ) ;
}

void PlayerDraw( int Co , int P1 , int P2 , int P3 )	// 自機の描画
{
	DrawLine( PlData.x2[P1],PlData.y2[P1],PlData.x2[P2],PlData.y2[P2],Sys.Cr[Co] ) ;
	DrawLine( PlData.x2[P2],PlData.y2[P2],PlData.x2[P3],PlData.y2[P3],Sys.Cr[Co] ) ;
	DrawLine( PlData.x2[P3],PlData.y2[P3],PlData.x2[P1],PlData.y2[P1],Sys.Cr[Co] ) ;
}

void WallDraw( int P1 , int P2 )	// 壁描画
{
	double y1 , y2 , y3 , y4 , v ;

	v = Sys.v[ P1 ] ;
	y1 = PlData.y - 300		; y2 = PlData.y + 300		;
	y3 = PlData.y + P2 - Sys.Wi ; y4 = PlData.y + P2 + Sys.Wi ;
	DrawBox((int)( 319 - 320 * v ),(int)( 199 + y1 * v ),
		(int)( 320 + 320 * v ),(int)( 200 + y3 * v ), Sys.Cr[2],TRUE ) ;
	DrawBox((int)( 319 - 320 * v ),(int)( 199 + y4 * v ),
		(int) (320 + 320 * v ),(int)( 200 + y2 * v ), Sys.Cr[2],TRUE ) ;
}

void WallDraw2( int P1 , int P2 )	// 壁描画２
{
	double y3 , y4 , v ;

	v = Sys.v[ P1 ] ; y3 = PlData.y + P2 - Sys.Wi ; y4 = PlData.y + P2 + Sys.Wi ;
	DrawBox( (int)( 319 - 320 * v ),(int)( 200 + y3 * v ),
		(int)( 320 + 320 * v ),(int)( 199 + y4 * v ), Sys.Cr[1] ,TRUE ) ;
}

void StateDraw( void )	// ステータス表示
{
	char StrBuf[ 16 ] ;
	int i ;

	// 枠描画
	for( i = 0 ; i < 3 ; i ++ )
		DrawBox( i , i , 16 * 12 - i , 16 * 6 - i , Sys.Cr[7] , FALSE ) ;

	// 文字描画
	DrawString( 2*16,   16 , "ＳＴＡＧＥ"      , Sys.Cr[7] ) ;
	DrawString( 2*16, 2*16 , "ＬＥＦＴ"        , Sys.Cr[7] ) ;
	DrawString( 2*16, 3*16 , "ＤＩＳＴＡＮＣＥ" , Sys.Cr[7] ) ;

	// メーター描画
	i = Sys.Di/(Sys.St+1) ;
	DrawBox( ( 3 + i ) *16, 4 *16,        10 *16 , 5 *16 , Sys.Cr[7] , TRUE ) ;
	DrawBox(          2 *16, 4 *16, ( 3 + i ) *16 , 5 *16 , Sys.Cr[2] , TRUE ) ;

	// 残機とステージ数描画
	StrConv( Sys.St + 1 , StrBuf ) ; DrawString( 8*16, 16  ,StrBuf,Sys.Cr[7] ) ;
	StrConv( Sys.Lf, StrBuf ) ;      DrawString( 8*16, 2*16,StrBuf,Sys.Cr[7] ) ;
}

void StrConv( int ConvData , char *DestStr )	// 数値を２バイト文字に変換
{
	int i , j ; unsigned short k ; char Str[ 100 ] ;

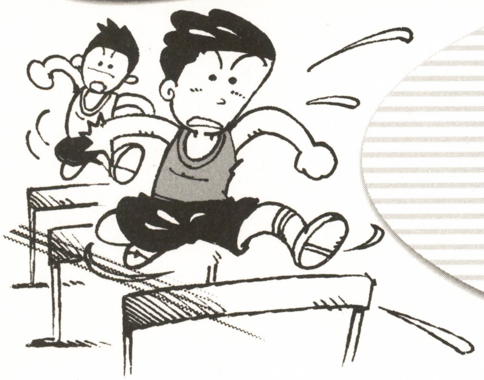
	itoa( ConvData,Str,10 ); j=lstrlen( Str );
	for( i = 0 ; i < j ; i ++ , DestStr += 2 ){
		k = _mbbtombc( ( unsigned short )Str[ i ] ) ; // ２バイト文字に変換
		DestStr[1] = (char)k ;				// 代入
		DestStr[0] = *(((char *)&k)+1) ;
	}
	*DestStr = 0 ;
}

int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance,	// WimMain関数
			LPSTR lpCmdLine, int nCmdShow )
{
	if( DxLib_Init() == -1 ) return -1 ;		// ＤｘＬｉｂの初期化
	SetUse3DFlag( FALSE ) ;				// ３Ｄ機能を使わない
	GameInit() ;					// ゲームの初期化
	TitleFunc() ;					// タイトル画面処理
	GameMain() ;					// ゲームメイン
	DxLib_End() ;					// ライブラリ終了
	return 0 ;					// 終了
}
```


★Java-SDKをダウンロードしていたら，途中で回線が切れた。16MBダウンロードに費やした1時間半を返して～（泣）。（青森県南津軽郡・ていこく22歳）……【Dr.D：ブロードバンド時代とはいえ，まだまだ全家庭が高速通信網に対応するのはまだ先じゃな。】

(超大作プログラム)　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(F-BASIC V6.0)用



陸上競技をマウスで体験

マウスde世界陸上

MONOMON（31歳）

はじめに

パチンコが指の運動から手首の運動に進化して久しいですが，そろそろTVゲームも指の運動から手首の運動に進化するころ(?)でしょう。

それでは，連打ゲームから進化した新しいスポーツ・ゲームをご堪能ください。

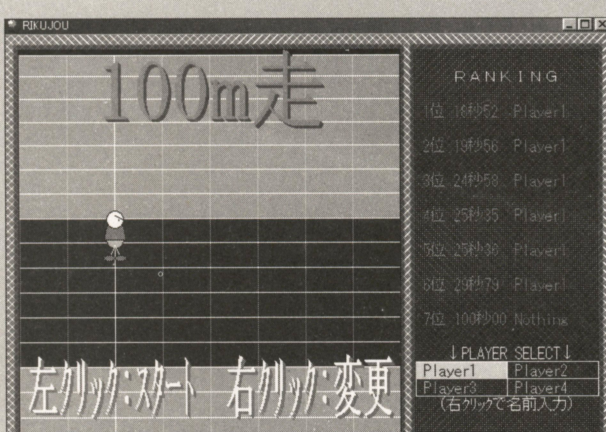
遊びかた

全部で６種目の競技で遊べます。右クリックで競技を切り替え，左クリックでスタートです。

画面右下のプレイヤーを選択してからプレイすると，記録更新時にその名前で保存されます。右クリックすると名前が入力できるので，みんなで記録を競い合いましょう。

競技説明

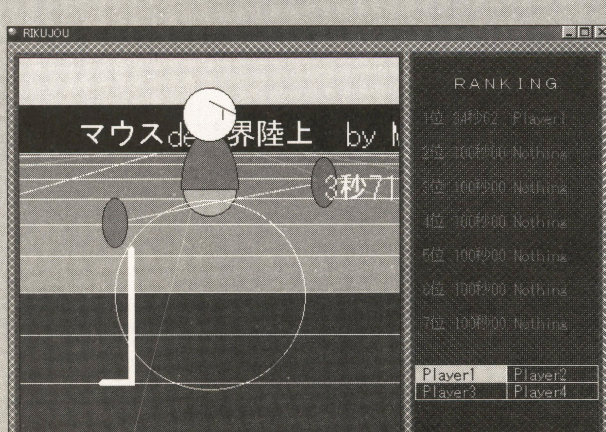
★100m走：ランナーの足元の円に合わせてマウスを回転させるとマンガ走りをします。円からはみ出すと足が短くなり，減速するので注意。



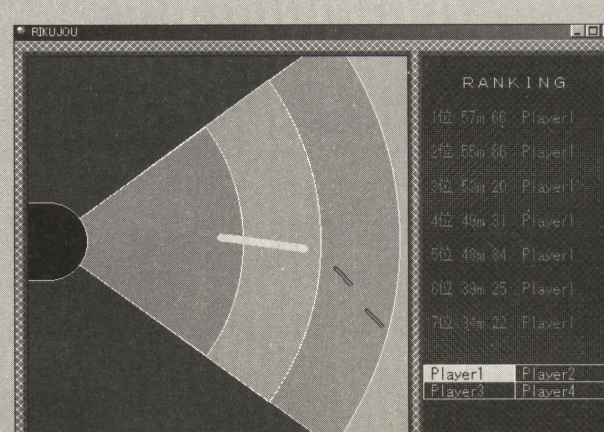
《写真１》マウスだけで遊べるゾ



《写真２》100m走の画面。マウスを回しまくれ



《写真３》ハードル走。ジャンプもしっかり再現



《写真４》槍投げなどの競技も用意されている

★プログラム投稿は，FDではなく，CD-ROM（CD-R）ではいけないのですか？（千葉県市原市・**増田武宣17歳**）……【**編：**もちろんOKです。CD-Rメディアも安くなりましたしね。FD，CD-Rどちらでもかまいませんが，1作品1枚のメディアに入れてください。】

★ハードル走：操作は100m走と同じです。ハードルが10mごとにあるのでジャンプしてください。マウスカーソルを，背景のフェンスぐらいの高さまで上げれば，ジャンプします（ハードルのないところではジャンプしません）。

★障害物走：操作は100m走と同じです。起伏に富んだコースを走るのでランナーがあちこちに動き回ります。円からはみ出すと減速するので頑張って追いかけてください。

★槍投げ：左クリックしながら，マウスをコースからはみ出さないようにすばやく動かし，離すと槍を投げます。なるべく右端ギリギリで離すと有利です。

★ハンマー投げ：左クリックしながら円内をグルグル回転し，離した方向へ投げます。なるべく多く回転させ，なるべく大きな円を描くように回したほうが距離が稼げますが，円からはみ出したり時間切れになるとファールとなります。

★ブーメラン投げ：操作はハンマー投げと同じです。ただ，回転数に応じてブーメランが曲がってしまいます。落下点とセンターの近さを%で表し，「距離×近さ」で出た得点が記録となる変わった競技です。

《第1表》変数表

・実験用定数	
WT=1	ゲームループの最小単位時間（大きい値を入れて処理が遅い場合の挙動を調査した）
L=200	表示画面サイズ（小さい値を入れ「画面外を描画しない」処理のチェックをした）
ランキング用変数	
SC(GMAX,RKMAX)	各競技の記録
SN$(GMAX,RKMAX)	記録保持者名
S1(GMAX,RKMAX)	記録補助変数1（走り競技時：最高速度，投げ競技時：着弾点のX座標）
S2(GMAX,RKMAX)	記録補助変数2（走り競技：倒したハードルの数と進行方向，投げ競技：着弾点のY座標）
・共通定数	
GMAX=5	競技の種類-1
RKMAX=7	最大ランキング数
PI2=2*3.1415926	2π(円周，角度計算用)
文字定数	
NM$(3)	競技者名簿
LOG$(7)	各種ロゴ（画面上に出る文字）
MES$(7)	各種コメント（画面下に出る文字）
SND$(9)	サウンドファイル名
GM$(GMAX)	競技名（タイトル時の見出し用）
・共通変数	
GF	ゲームフラグ（0:100m走，1:ハードル走，2:障害物走，3:槍投げ，4:ハンマー投げ，5:ブーメラン投げ）
RK	プレイヤーのランク（0-6:ランクイン，7:選外）
PL	競技者名簿選択番号
MF	メッセージフラフラグ（LOG$(MF)MES$(MF)で各種文字列を取得）
X,Y,Z	画面表示座標（X，Yは倍率1の時m＝1ドット，Zは倍率で値は1-20で指定）
X0	カメラ接近演出用初期画面表示X座標
MOGX,MOGY	モグラ出現座標
GT	GetTickCount時間保存用変数
DT	単位時間（プログラム1ループの時間をミリ秒単位で取得）
T	ローカルタイマー（そのつどリセットし必要な時間をカウント）
A$,B$,A,B,C,I,J	雑用
・マウス操作カウント変数	
MX,MY	マウス座標
MX0,MY0	旧マウス座標
CX,CY	マウス回転操作の中心点座標
N	マウスカウントルーチン呼び出し回数（要素数として累積データを割ると平均値が出る）
MV	マウスカーソル総移動距離（累積値）
R	半径（CX，CYからのマウスの距離）
R0	旧半径
TR	半径の累積値
D	マウス方向（CX，CYから見た角度）
D0	旧マウス方向
DRT	回転角度（DとD0の差分）
ROT	総回転数（DRTの累積値）
・競技別変数	
W	フィールド幅（走り競技時はトラック距離[単位:m]，投げ競技時は扇の角度[25で180度=最大値]）
V	速度（走り競技時はランナーの，投げ競技時は投げ物体の速度）
MANF	走り競技・ランナー表示フラグ（＝高さ）
AX1,AY1,AX2,AY2	走り競技・ランナーの足位置座標
D0,D1	投げ競技・投げ方向（初期角度，角度変化速度）
HT	投げ競技・投げ物体の滞空時間
HH	投げ競技・投げ物体の最高高度（Z:倍率と同じ単位で指定）
HDL(10)	ハードル変数（1:ハードルあり，0-1:ハードル倒れ中，0以下:ハードルなし）
JAM(20,1)	障害物変数（引数1:5mおきの位置，引数2: 0=高さ，1=傾斜 の情報）
・各種オブジェクト変数	
BUF	ウラ画面用ビットマップオブジェクト
FORM	フォームオブジェクト

サウンドについて

今回，ためしに変わったことをしてみました。今までの作品ではWAVファイルを別途用意してもらってましたが，Program FilesフォルダのWAVファイルを直接呼び出してみました。

ただ，そのファイルがインストールされていないこともあると思うので，初期起動時にSOUND.TXTというファイルを作成し，その中身をユーザが書き換えることで自由に音を変更できるようにしています。

こんなやり方って市販ソフトじゃ反則でしょうが，個人が作って楽しむプログラムではお手軽でいいんじゃないでしょうか（友人に配布するときも楽だし）…ってダメ？

苦労したところ

処理速度の確保のため，画面外を極力描画しないようにしたのですが，そこの計算は苦労しました。

3Dっぽく作ってはいても，全然3Dじゃないので計算が一様化できず，かえって複雑になったようです。

それでも，計算が合ってれば画面が表示されるので，チェックしやすいという点では楽だったかもしれません。

ということで最も苦労した点は，走りの速度＆減速計算や投げの速度＆滞空時間の計算です。記録に差がつき，しかも納得でき，面白いと呼べる調整ができたかは，正解がなくて大変でした。一応納得いくまで調整したつもりですが，どうでしょう皆さん？って感じですね。

最後に

元々はハンマー投げのみだったのですが，すぐファールになってストレスがたまるうえに，プレイ時間が速すぎて「遊んだ」という満足感が得られにくいという不満点がありました（唯一満足感が得られる「新記録達成」もプレイ回数が増えると取りにくくなっていくし…）。

で，プレイ時間もそこそこあって「ハイスコアは取れなくても1面クリアした」みたいな満足感が得られるものをと，100m走を追加したのです。もちろんストレスをなくすため，フライングもなしにしました。

その結果，リストも長く汚いものになってしまいましたが，さらに競技を増やして遊べるものになったと思います。でも，今回はかなり苦労したわりに飽きが早いかもしれません。

☆　☆　☆

CHECKER FLAG

影：これは，かなり良く作られている。マウスというデバイスの特徴をうまく使って，この手のゲームには必須条件と言えるプレイヤー操作重視のプレイ形式をうまく再現しているゾ。

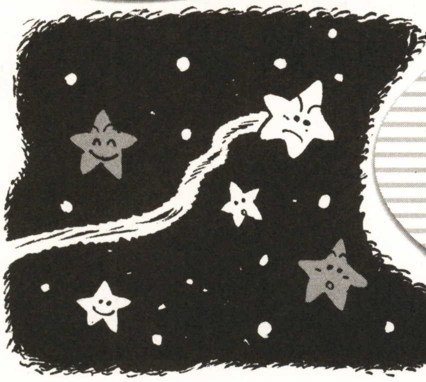
つぐ美：グラフィックはチョット貧弱に見えるかもしれないけど，スタート前にプレイヤーの場所まで拡大したりなど，演出面がしっかり作られているわ。

編：市販の同じジャンルのゲームと比べても，熱中度という面では負けていない作品です。

Dr.D：若干リストが長めだったので，超大作での掲載じゃが，ぜひともリストにも注目してほしい作品じゃ。

★誕生日迎えました。どんどん，つぐ美さんより年上に…。（福島市・**南斗ライバ25歳**）……**【つぐ美：**おめでとう！私より5歳も年上になったのね。で，プレゼントは何かもらったの？**影：**最近読者から何ももらえないつぐ美ちゃんは，プレゼントにナーバスです。**】**

Windows(HSP V2.5)用



スピード感のある障害よけ

流(なが)れ☆(ほし)

@SYD（16歳）

内容

言ってみれば，ただの障害物よけゲームです。前回のプログラム(2001年5月号掲載，HSP用『@MAZE』)がとても長かったので，今回は短めにしてみましたが，どうでしょうか。



《写真》慣性のついた動きをうまく制御

操作方法

カーソル・キーで操作します。自機が画面外に出たときか，または障害物に当たったときは，ゲーム・オーバーです。タイムが0になるまで逃げ切ったらゲーム・クリアとなります。なお，できるだけ画面の右側にいたほうがスコアが上がります。また，14～16行を注釈の命令に変更するとマウス操作になります。

その他

今回は短いプログラムを目指しているので省略してますが，グラフィックを変えたり，ライフ制にしたり，ハイスコアセーブ機能をつけたりいろいろできると思います。

私の最高得点は100万点です。多分このぐらいではすぐに抜かれてしまうでしょう。ご意見，ご感想などありましたらcyh04531@nifty.comまで。

《第1表》変数表

PX	プレイヤーX座標
PY	プレイヤーY座標
MX	X軸方向移動距離
MY	Y軸方向移動距離
SC	スコア
TM	残りタイム

CHECKER FLAG

編：この作品の特徴は，やはりリストの短さ。この手の作品につき物の，当たり判定を行うif文が一つしか見あたらないのですが，Dr.これはどういう仕組みなんですか？

Dr.D：この作品は，HSPの命令の特徴をうまく利用しているのが，リストの縮小につながっておるのじゃ。編クンの疑問点である星との当たり判定部分は，“pget”命令で指定した座標の色データを得て，その色データ（RGB輝度で示される）の一つであるbvalの値を使って，唯一のif文で当たり判定を行っておる。

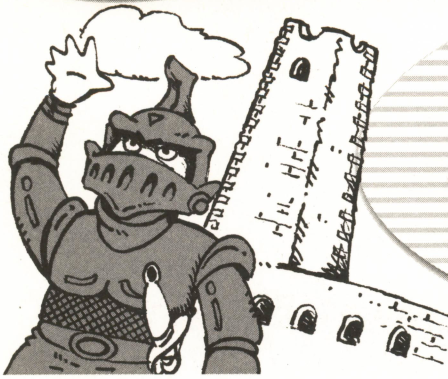
つぐ美：ゲームとしては，スピード感があるゲームに仕上がってますよね。

『流れ☆』のプログラム・リスト

```
gsel 0,-1 :chgdisp 3 :bgscr 2,640,480,1 :mouse -1 :gmode 1          ;初期化
randomize :px=16 :py=232 :mx=0 :my=0 :sc=0 :tm=0
repeat 16                                                    ;☆のパレット設定
        palette cnt,cnt*8+127,cnt*8+127
        palette -cnt+31,cnt*8+127,cnt*8+127
loop                                                         ;★のパレット設定
repeat 128
        rnd r,256 :rnd b,256 :palette cnt+32,r,255,b
loop
palette 160 :palfade :palcolor 160 :boxf                          ;背景を塗る
palcolor 255 :pos 200,0 :mes "- 流れ☆ (C) @ＳＹＤ 2001 -"
repeat
        redraw 2
        stick key,15                       ;
        mx+(key&4/4)-(key&1)    :px+mx    ;px=mousex
        my+(key&8/8)-(key&2/2) :py+my     ;py=mousey
        tm=3999-cnt :sc+px :rnd y,-cnt+4000                    ;タイム&スコア
        palcolor 160 :boxf 632                                  ;背景を塗る
        font "ＭＳ ゴシック",32 :palcolor cnt¥128+32:pos 608,y :mes "★"
        font "ＭＳ ゴシック",16 :palcolor cnt¥32    :pos px,py :mes "☆"
        pget px+8,py+8  :pos 0,16          :gcopy 2,8,16,632,464;スクロール
        palcolor 160 :boxf 0,0,192,15                          ;↓の背景クリア
        palcolor 255 :pos 0,0 :mes "TIME:"+tm+" SCORE:"+sc            ;表示
        redraw 1,0,0,608,480
        if px<8|(px>608)|(py<8)|(py>464)|(tm=0)|(bval!0) :break
        await 16
loop
palcolor 255 :pos 240,232                                     ;ゲーム終わり
if tm :mes "GAME OVER :"+sc :else :mes "GAME CLEAR :"+sc
wait 500
```


★ウチの中学校のパソコン室には，なぜか「テラドライブ」の取り説だけあります。ある意味“七不思議”。（東京都練馬区・**ルストフレム14歳**）……【**影：**七つの不思議な事柄があって七不思議というんだよ。ということで，あとの六つも知りたいものだ。よろしく。】

Windows(QBasic)用



人気の投稿作品，今度はQBasicへ

移植版

聖騎士(せいきし)と魔王(まおう)の塔(とう)

tayosan（17歳）

はじめに

このゲームは，ベーマガ1997年４月号に掲載された樫部雄介さんの作品をジャイアニズムさんがHSPに移植して，それを僕がQBasicに移植したものです。原作およびステージ(6-15)のアイデアは樫部雄介さんです。それとこれは，HSPの作品をもとに作っています。

HSP版との違い

見た目が違う。アルゴリズムが違う。操作方法が違う。アルゴリズムを変えたので，結構短くなったかな，と思います。それと超初心者用の面を六つプラスしました。とてつもなく簡単です。

遊びかた

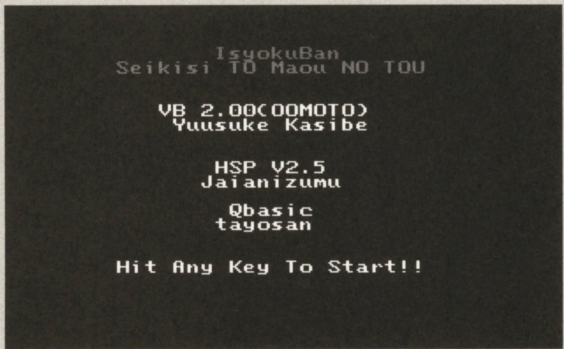
実行してキーを押すとステージセレクトができますので，やりたい面を選んでください。１面からやるならそのまま エンター を押してもかまいません。騎士は“！”，“％”がタイルで，このタイルの上に騎士が来ると“＊”になって進めなくなります。動かしかたはチェスのナイトと同じ八方桂です。動ける所には数字が出ているので行きたい所の数字押してください。タイルのすべてを“＊”にすればクリア，エンター キーでリトライ，Esc キーでタイトル（タイトル中に押すと終了）です。ゲーム・オーバーになったら面セレクトで始めてください。

苦労したこと

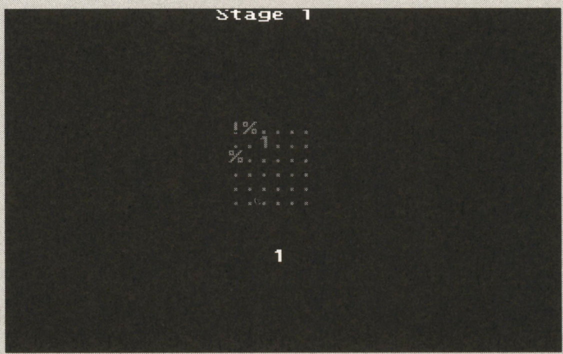
なし。あ，HSP のデータ解析が…。初期位置はどうやって決めてんの？と言う感じで…。結局わからんかった（HSPもってな～い）ので初期位置を２にした。このせいでクリアできない面があるかも…。その時はお知らせください…。

改造のアドバイス

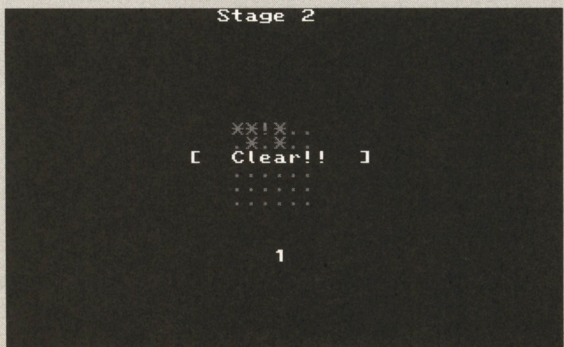
面の追加は簡単だと思うので。それと



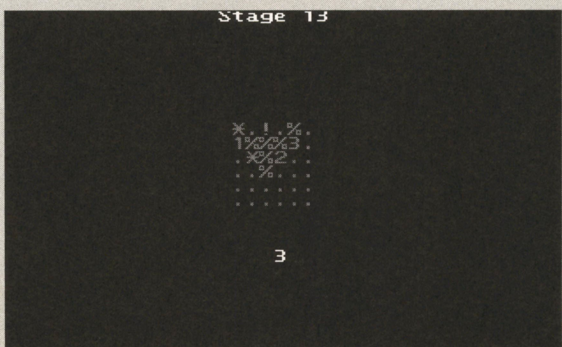
《写真1》QBasic版の『聖騎士と魔王の塔』



《写真2》コマを%の場所へ移動させていく



《写真3》序盤のステージで内容を把握しよう



《写真4》移動先を数字で選択できるのが魅力

CHECKER FLAG

Dr.D：リストの面で見ると，ステージを表す配列とキャラクターの移動先を求める処理さえできれば，どの言語であろうとほぼゲーム内容を再現できるのが特徴じゃな。

つぐ美：このQBasic版の特徴は，駒の移動先を数字で表して，その数字を入力することで移動できる点ね。先月PC98版のこの作品をチェックしたときは，マウスを使わないと遊びにくい作品かな，という印象を持ったのですが，その部分をアイデアでカバーしています。

このしょぼい画面を派手にするとかやってください。

最後に

感想がありましたらtayosan@tokutoku.zzn.comまで送ってくださ〜い!!

☆　☆　☆

（移植版）『聖騎士と魔王の塔』のプログラム・リスト

```
DECLARE SUB Msg (St$, Py!, Cl!)
smax = 14
DIM Ban(5, 5), Map(5, 5, smax), sx(smax), sy(smax), MoveX(7), MoveY(7), MoveOk(7)
DIM ch$(11), Td$(8), Ty(8), Tc(8), i$, a$, f$, kt1, kt2, mj, mi

SCREEN 13: CLS

FOR i = 0 TO 3: READ ch$(i): NEXT
FOR i = 1 TO 8: ch$(3 + i) = MID$(STR$(i), 2): NEXT
FOR i = 0 TO 7: READ MoveX(i), MoveY(i): NEXT
FOR i = 0 TO smax: READ y: FOR j = 0 TO y - 1: READ a$: FOR k = 0 TO 5
Map(k, j, i) = VAL(MID$(a$, k + 1, 1))
IF Map(k, j, i) = 2 THEN sx(i) = k: sy(i) = j
NEXT k, j, i
FOR i = 0 TO 8: READ Td$(i), Ty(i), Tc(i): NEXT

title:
CLS : FOR i = 0 TO 8: Msg Td$(i), Ty(i), Tc(i): NEXT
PLAY "o3l32t255cbcbdadagegefefeefefdgdgcaca"
GOSUB keyin: IF i$ = CHR$(27) THEN PLAY "a5g4f4e3d2c1": RUN "qbmenu"''END"

first:
PLAY "o4l32t255cdf"
CLS : Msg "Input Stage ( 1 -" + STR$(smax + 1) + " ) >", 5, 4: INPUT s
IF s = 0 THEN s = 1 ELSE IF s < 1 OR s > smax + 1 THEN GOTO first
sec: CLS : IF s > smax + 1 THEN GOTO all
Msg "Stage" + STR$(s), 1, 15: x = sx(s - 1): y = sy(s - 1)
FOR i = 0 TO 5: FOR j = 0 TO 5: Ban(j, i) = Map(j, i, s - 1): NEXT j, i
FOR i = 3 TO 6: PLAY "o" + MID$(STR$(i), 2) + "t140l40cdfb": NEXT
GOSUB MapPut: GOTO main

MapPut:
FOR mi = 0 TO 5
  FOR mj = 0 TO 5
    LOCATE 9 + mi, 17 + mj
    COLOR Ban(mj, mi) + 1
    PRINT ch$(Ban(mj, mi))
  NEXT mj
NEXT mi
RETURN

main:
GOSUB mvht: LOCATE 9 + y, 17 + x: COLOR 3: PRINT ch$(2)
GOSUB keyin: nb = VAL(i$): IF i$ = CHR$(27) THEN GOTO title
IF i$ = CHR$(13) THEN GOTO sec
IF nb > 0 AND nb < kt1 + 1 THEN
  Ban(x, y) = 3: x = x + MoveX(MoveOk(nb - 1)): y = y + MoveY(MoveOk(nb - 1))
  Ban(x, y) = 2: GOSUB MapPut: PLAY "l16t120o3b>dbd<b"
END IF
GOTO main

keyin:
WHILE INKEY$ <> "": WEND: i$ = ""
WHILE i$ = "": i$ = INKEY$: WEND: RETURN

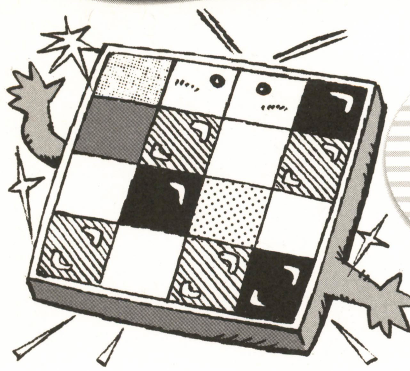
mvht:
kt1 = 0: kt2 = 0
FOR i = 0 TO 7: lx = MoveX(i) + x: ly = MoveY(i) + y
  IF lx > -1 AND lx < 6 AND ly > -1 AND ly < 6 THEN
    IF Ban(lx, ly) = 1 THEN
      MoveOk(kt1) = i: kt1 = kt1 + 1
    END IF
  END IF
NEXT i
FOR i = 0 TO 5: FOR j = 0 TO 5
  IF Ban(i, j) = 1 THEN kt2 = kt2 + 1
NEXT j, i
IF kt2 = 0 THEN GOTO Cle
IF kt1 = 0 THEN GOTO Ove
FOR i = 0 TO 1
  FOR j = 0 TO kt1 - 1: Msg STR$(kt1), 18, 15
    IF i = 0 THEN nb = j + 4 ELSE nb = 1
    Ban(x + MoveX(MoveOk(j)), y + MoveY(MoveOk(j))) = nb
  NEXT j
  IF i = 0 THEN GOSUB MapPut
NEXT i
RETURN

Cle: PLAY "o3": FOR i = 1 TO 2: PLAY ">" + "l64t255cc+dd+eff+gg+aa+b": NEXT
Msg "[  Clear!!  ]", 11, 12: GOSUB keyin: s = s + 1: GOTO sec
Ove: PLAY "o4T120l4gfedc"
Msg "[  Oh No!!  ]", 11, 11: GOSUB keyin: GOTO title
all: PLAY "l20t120o4cdfgfa2"
Msg "[  Stage All Clear!!  ]", 11, 13: GOSUB keyin: GOTO title

DATA .,%,!,*,-1,-2,-2,-1,-2,1,-1,2,1,2,2,1,2,-1,1,-2
DATA 3,21,001,1,3,2111,0101,0111,3,10201,001,0101,4,211,011,1111,1,4,211,0101
DATA 1111,101,3,1111,011,1111,3,2111,1111,1111,4,211,111,111,111,4,11111
DATA 11111,11111,11111,5,2111,1111,1111,1111,1111,5,11111,11111,11111,11111
DATA 11111,4,211111,111111,111111,111111,4,20101,11111,0111,001,4,2111,01111
DATA 1111,01111,6,02111,111111,101101,111111,01111,0011
DATA IsyokuBan,4,5,Seikisi TO Maou NO TOU,5,2,VB 2.00(OOMOTO),8,15
DATA Yuusuke Kasibe,9,12,HSP V2.5,12,15,Jaianizumu,13,13,Qbasic,15,15
DATA tayosan,16,11,Hit Any Key To Start!!,19,15

SUB Msg (St$, Py, Cl)
        LOCATE Py, 20 - LEN(St$) / 2
        COLOR Cl: PRINT St$;
END SUB
```


★パソコン自作しマース！…って，そんなことベーマガに報告したら……破壊魔参上！（鹿児島県加世田市・IWACHAN15歳）……
【影：と，サングラスをかけた男の絵が描いてありました。いったいこれ誰？私は破壊魔を卒業しましたし…。その絵がこれ⇒ 】



iアプリで手軽にプレイ

BLACK LIGHT（ブラック ライト）

タカヒ（27歳）

内容

押すと色の変わるパネルを使って，すべての色を同じにするゲームです。かなり昔に，こんなおもちゃがありましたよね？

操作方法

上下左右でカーソルを動かします。セレクトボタンを押すと，カーソルのあるマスを中心に3×3マスの色が反転します（黄色⇔青）。これを繰り返すことですべてのパネルを同じ色にできれば「おめでとう」と言われます。

ソフトキー2（右側のソフトキー）を押すとNEWゲームとなり，違うパターンから始められます。ソフトキー1でプログラムが終了します。

プログラムの仕組み

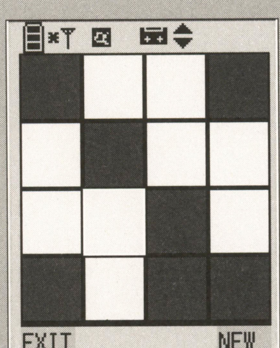
キー入力がされるとパネルのパターンを構成して再描画する。その繰り返しで成り立っているだけの単純な物です。

プログラムの入力法と注意点

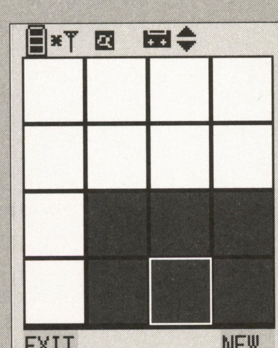
メソッドで使っている変数xとyは，パネル配列(Pane[][])のY軸，X軸の添え字として統一しています。各メソッドで同じ使いかたをしているので，勘違いさえしなければ問題ないと思います。ちなみにカーソル位置の変数はcx，cyです。

改造のアドバイス

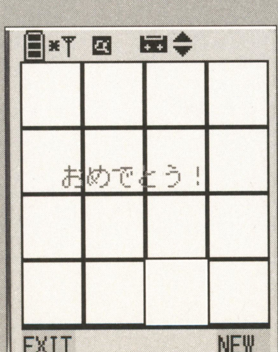
パネル配列(Pane[][])と，パネル数を表す変数siz。この二つを変えることで，



《写真1》定番ゲームの一つなので遊びやすい



《写真2》反転の性質をしっかり覚えよう



《写真3》すべて点灯さえれば完成だ

CHECKER FLAG

つぐ美：定番ゲームの一つ。内容は単純なもので新鮮味に欠けますが，iアプリプログラムの入門という見かたでは，参考になる作品かもしれませんね。

Dr.D：iアプリ投稿が増えているのは良い傾向じゃが，思っていたよりこだわった作品が少ないという印象じゃ。確かに現在のiアプリの10Kバイト制限は厳しいかもしれんが，逆に言えば作品のアイデアがより重視される環境と言える。こういう環境のほうがアマチュアプログラマーにとっては，腕の見せ所なのではないかな？これからの投稿に期待したい。

編：また，これはよくある投稿不備なのですが，作品タイトルは説明ファイルにしっかりと記述しておいてください。今回はファイル名などから「BLACK LIGHT」というタイトルと判断できましたが，作品によっては何の手がかりもない場合があり，そのためにボツになったりしますので。

★copyrightを“版権”と訳した人は誰でしょう。………正解：福沢諭吉（ちょっと勉強になったでしょ？）。（兵庫県神戸市・JANDY18歳）……【つぐ美：うん，たしかに勉強になったわ。ついでに，trademark（トレードマーク）を商標と訳した人も調べておいて。】

パネルの全体数を増減できます。ただ，パネル数を増やしても難易度はそれほど変化しないようです。面倒な作業が増えるだけといった感じでしょうか？

苦労したところ

開始前にパネル数を選択できるようにしようとか，完成時に奇麗に点滅させようとか思ったのですが，ベーマガ的に短く，簡潔な内容で終わらせました（手抜きとも言う）。

最後に

N88-BASICでゲームを作っていた学生時代から，約10年ぶりにベーマガを購入しました。当時の遅いパソコンと，現在の携帯iモードとは性能面でも似たような状況であり，開発意欲をそそります。

ですが10年の変化はなかなか受け入れ難く，N88-BASICとクラス処理をメインにしているJavaとの差に戸惑っている状況です。そんな中で作ったプログラムなのでひどい物ですが，これからも頑張っていきたいと思います。当然，ベーマガも毎月購入しますよ（笑）！

《リスト1》Black Light.javaのプログラム・リスト

```
import com.nttdocomo.ui.*;
import java.util.Random;

public class BlackLight extends IApplication {
        MainCanvas gc;

        // スタートメソッド
        public void start() {
                gc = new MainCanvas();
                Display.setCurrent(gc);
        }
}

class MainCanvas extends Canvas {

        int px = 0;                        // パネルのX，Y設定
        int py = 0;
        int Pane[][] = new int[4][4];      // パネルの配列
        int cx = 0;                        // カーソルのX，Y設定
        int cy = 0;
        int ss = 0;                        // セレクトボタン判定
        int siz = 4;                       // パネルサイズ設定
        int nan = siz * 5;                 // パネルのランダム変換回数
        int con = 0;                       // 終了判定
        int kai = 0;                       // 変換回数（セレクトの回数）

        // ＭａｉｎＣａｎｖａｓのコンストラクタ
        public MainCanvas() {
                setSoftLabel(Frame.SOFT_KEY_1, "EXIT");
                setSoftLabel(Frame.SOFT_KEY_2, "NEW");
                px = getWidth() / siz;
                py = getHeight() / siz;
                for(int x = 0; x < siz; x++)
                        for(int y = 0; y < siz; y++)
                                Pane[y][x] = 0;
                syoki();

        }

        // 描画処理
        public void paint(Graphics g) {

                g.lock();
                g.setColor(Graphics.getColorOfName(Graphics.BLACK));
                g.fillRect(0, 0, getWidth(), getHeight());

                // パネル描画
                kai = 0;                           // 判定リセット
                for(int x = 0; x < siz; x++){
                        for(int y = 0; y < siz; y++) {
                                if(Pane[y][x] == 0){
                                    g.setColor(Graphics.getColorOfName(Graphics.YELLOW));
                                    kai++;
                                }
                                else {
                                    g.setColor(Graphics.getColorOfName(Graphics.BLUE));
                                }
                                g.fillRect(x * px + 1, y * py + 1, px -2, py - 2);

                                }
                        }

                // 完成？
                if (kai == 0 || kai == siz * siz) con =1;

                // 完成おめでとう！
                if(con == 1) {
                        g.setColor(Graphics.getColorOfName(Graphics.RED));
                        g.drawString("おめでとう！", 20, 60);
                }
```


★高3です。専門学校に行くための勉強をしています（C言語など）。できたら投稿したいと思ってるので，そのときはよろしく！（石川県金沢市・**紙田　翼18歳**）……**【編**：受験がんばってね。**影**：ベーマガのキャラクター使うと採用率高いらしいよ（ひそひそ）。**】**

```
        //  カーソル描画
        g.setColor(Graphics.getColorOfName(Graphics.WHITE));
        g.drawRect(cx * px, cy * py, px -1, py - 1);
        g.unlock(true);

}

//  NEWGAME
public void syoki(){
        int x;
        int y;
        Random rand = new Random();
        int mr = Math.abs(rand.nextInt()) % nan;
        for(int h = 0; h < nan; h++) {
        x = Math.abs(rand.nextInt()) % siz;
        y = Math.abs(rand.nextInt()) % siz;
        henkan(x, y);
        }
}

//  3＊3マス変換
public void henkan(int xx, int yy){
        int minX = xx - 1;
        int maxX = xx + 1;
        int minY = yy - 1;
        int maxY = yy + 1;
        if (xx == 0) {
                minX = 0;
                maxX = xx + 1;
        }
        if (xx == siz-1) {
                maxX = xx;
                minX = xx - 1;
        }
        if (yy == 0) {
                minY = 0;
                maxY = yy + 1;
        }
        if (yy == siz-1) {
                maxY = yy;
                minY = yy - 1;
        }
        for(int x = minX; x <= maxX; x++)
                for(int y = minY; y <= maxY; y++)
                        if(Pane[y][x] == 1) Pane[y][x] = 0;
                        else Pane[y][x] = 1;
}

//  キー入力判定
public void processEvent(int type, int param) {
        if (type == Display.KEY_PRESSED_EVENT) {
                con = 0;
                switch(param) {
                        case Display.KEY_SOFT1:
                                IApplication.getCurrentApp().terminate();
                                break;
                        case Display.KEY_UP:
                                if(cy > 0)
                                cy--;
                                break;
                        case Display.KEY_DOWN:
                                if(cy < siz - 1)
                                cy++;
                                break;
                        case Display.KEY_LEFT:
                                if(cx > 0)
                                cx--;
                                break;
                        case Display.KEY_RIGHT:
                                if(cx < siz - 1)
                                cx++;
                                break;
                        case Display.KEY_SELECT:
                                ss = 1;
                                break;
                        case Display.KEY_SOFT2:
                                syoki();
                                break;
                        default:
                                break;
                }
        }
  //  セレクトが押されていれば変換処理へ
  if (ss == 1) henkan(cx, cy);
  ss = 0;

  //  再描画処理
  repaint();
  }
}
```

《リスト2》BlackLight.jamのプログラム・リスト

```
AppName = BlackLight
PackageURL = BlackLight.jar
AppSize = 1923
AppClass = BlackLight
LastModified = Fri, 27 Apr 2001 11:39:25
```

iアプリ

★6月号136ページ左下の絵の女の子の名前は，“ラ・ビ・アン・ローズ”と正確にはいうみたいです。（愛知県西尾市・ごぎん31歳）……【影：貴重な情報ありがとう。つぐ美：あら？影さんって，デジキャラのファンでした？影：いや，ファンにこの情報を売るのさ。】

ワンダースワン(WonderWitch)用



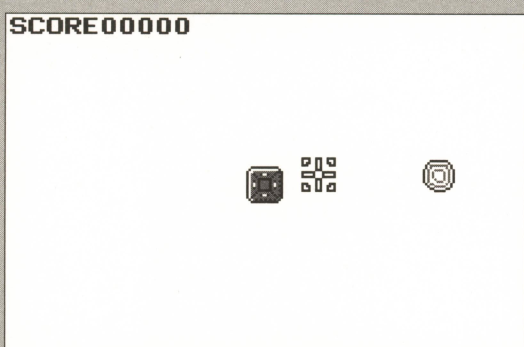
ゼロタンク再び発進！！

ゼロタンク２

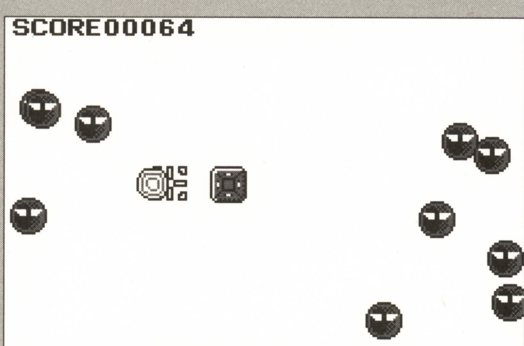
たくちゃん（20歳）

内容とストーリー

タンクを操って敵を破壊するゲームです。舞台は西暦XXX1年，異星人に侵略された地球。こんどこそ最後のヘッポコ兵器「ゼロタンク２」が発進した。もう玉砕は間近だ！



《写真１》今度はゼロタンク２が出動



《写真２》回転移動がこのゲームの特徴



《写真３》敵に照準をうまく合わせていこう

操作方法

左右のキーで戦車の向きを回転させます。上キーで戦車を前進させられます。弾はボタンを押さないでも連射し続けます。敵は全方向から襲ってきます。弾で倒すとその数がカウントされ，だんだん難易度が上がっていきます。

プログラムの仕組み

このプログラムは『ゼロタンク』(01年７月号)を改造して，自機の向きが回転するようにしたものです。全方向分のベクトルを記した配列を用意して，“sin”“cos”関数の代わりに使っています。

プログラムの入力法と注意点

makefile”を呼び出す都合上，“sample”フォルダの下にフォルダを作ってコンパイルしてください。

CHECKER FLAG

影：先月号掲載の『ゼロタンク』の続編。回転する移動方法が特徴の作品だ。

Dr.D：前作と同様に，自分で定義した各角度に対応した移動量を定義しているベクトルテーブルを利用しているのが，リストの部分では目につくのぉ。

つぐ美：動作速度の面や処理方法を考えると難しいけど，背景を作成して，戦車回転に合わせて背景が動くようになると，かなり見栄えのする作品になりそうですね。

★近所の書店では，ベーマガが1冊しかないので，毎月スリル満点です。P.S.6月号23ページのDr.Jに注目！…ヅラ…！！（千葉県銚子市・BrueJack17歳）……【編：どこの書店かお知らせください。販売部長に報告しておきます。影：スリル感がなくなるじゃん。】

《リスト１》zerot2.cのプログラム・リスト

```
/*      ゼロタンクII          */

#include <stdio.h>
#include <sys/bios.h>
#include <sys/fcntl.h>
#include <stdlib.h>

#define TEKI_MAX  12     /*敵の最大数*/
#define CHR_PIX 8            /* 1キャラクタの1辺の長さ[ピクセル]*/
#define CHR_BYTE 16      /* 1キャラクタあたりデータ数[バイト](4階調時) */

#define CPN_SP 8         /*キャラクタパレット番号：スプライト範囲*/
#define CPN_MY 12        /*キャラクタパレット番号：兼用*/
#define CPN_TEXT 0

#define MAS_PIX 16       /* 1マスの一辺[ピクセル数] */
#define MAS_CX  (MAS_PIX / CHR_PIX)      /* 1マスあたりの一辺[キャラクタ数] */
#define MAS_CHR (MAS_CX * MAS_CX)        /* 1マスあたりの総キャラクタ数 */

#define CHN_START 0                                  /*キャラクタ番号:*/
#define CHN_MY      CHN_START                    /*キャラクタ番号:主人公*/
#define CHN_TAMA   (CHN_MY    + MAS_CHR)  /*キャラクタ番号:弾*/
#define CHN_SCOPE  (CHN_TAMA + MAS_CHR)   /*キャラクタ番号:弾*/
#define CHN_TEKI   (CHN_SCOPE+ MAS_CHR)   /*キャラクタ番号:敵*/
#define CHN_NONE   (CHN_TEKI + MAS_CHR)   /*キャラクタ番号:空白*/
#define CHN_TEXT   (CHN_NONE + 1)         /*キャラクタ番号:テキスト*/
#define CHN_TEXT2  (CHN_TEXT + 10)        /*キャラクタ番号:テキスト2*/

#define SPN_ST     0                              /*スプライト番号:最初*/
#define SPN_MY     SPN_ST                     /*スプライト番号:主人公*/
#define SPN_TAMA   (SPN_MY    + MAS_CHR)  /*スプライト番号:弾*/
#define SPN_SCOPE  (SPN_TAMA + MAS_CHR)   /*スプライト番号:スコープ*/
#define SPN_TEKI   (SPN_SCOPE+ MAS_CHR)   /*スプライト番号:敵*/
#define SP_MAX     (SPN_TEKI + MAS_CHR*TEKI_MAX) /*スプライト数  :最大使用数*/

#define OBJ_NONE 255

/*----------------ベクトル用構造体*/
typedef struct {
    int  dx;    /* X座標ベクトル */
    int  dy;    /* Y座標ベクトル */
} VECT;

#define ANG_MAX 36

VECT vec[ANG_MAX] = {
 {   0 ,-100},  /*   0° */
 {  17, -98},   /*  10° */
 {  34, -93},   /*  20° */
 {  49, -86},   /*  30° */
 {  64, -76},   /*  40° */
 {  76, -64},   /*  50° */
 {  86, -50},   /*  60° */
 {  93, -34},   /*  70° */
 {  98, -17},   /*  80° */
 { 100,    0},  /*  90° */
 {  98,  17},   /* 100° */
 {  93,  34},   /* 110° */
 {  86,  49},   /* 120° */
 {  76,  64},   /* 130° */
 {  64,  76},   /* 140° */
 {  50,  86},   /* 150° */
 {  34,  93},   /* 160° */
 {  17,  98},   /* 170° */
 {   0, 100},   /* 180° */
 { -17,  98},   /* 190° */
 { -34,  93},   /* 200° */
 { -49,  86},   /* 210° */
 { -64,  76},   /* 220° */
 { -76,  64},   /* 230° */
 { -86,  50},   /* 240° */
 { -93,  34},   /* 250° */
 { -98,  17},   /* 260° */
 {-100,    0},  /* 270° */
 { -98, -17},   /* 280° */
 { -93, -34},   /* 290° */
 { -86, -49},   /* 300° */
 { -76, -64},   /* 310° */
 { -64, -76},   /* 320° */
 { -50, -86},   /* 330° */
 { -34, -93},   /* 340° */
 { -17, -98},   /* 350° */
};

/*----------------キャラデータ*/
char cChrDat[]={
  /*主人公 (戦車) */
 0x00, 0x3F, 0x3F, 0x7F, 0x7F, 0xE0, 0x6D, 0xD3, 0x77, 0xC8, 0x78, 0xC7, 0x6B, 0xD4, 0x7B, 0xD4
, 0x00, 0xFC, 0xFC, 0xFE, 0xF8, 0x07, 0xB6, 0xC9, 0xEC, 0x13, 0x1E, 0xE1, 0xD4, 0x2B, 0xDE, 0x29
, 0x7B, 0xD4, 0x6B, 0xD4, 0x78, 0xC7, 0x77, 0xC8, 0x6D, 0xD3, 0x5F, 0xE0, 0x35, 0x6A, 0x00, 0x3F
, 0xDC, 0x2B, 0xD6, 0x29, 0x1C, 0xE3, 0xEE, 0x11, 0xB4, 0xCB, 0xFA, 0x05, 0x54, 0xAA, 0x00, 0xFC

  /*弾*/
, 0x00, 0x00, 0x00, 0x0F, 0x0F, 0x1F, 0x1F, 0x30, 0x1F, 0x37, 0x3F, 0x6C, 0x3F, 0x6B, 0x3F, 0x6B
, 0x00, 0x00, 0x00, 0xF0, 0xF0, 0xF8, 0xF8, 0x0C, 0xF8, 0xEC, 0xFC, 0x36, 0xFC, 0xD6, 0xFC, 0xD6
, 0x3F, 0x6B, 0x3F, 0x6D, 0x3F, 0x6E, 0x1F, 0x33, 0x0F, 0x1C, 0x03, 0x0F, 0x00, 0x03, 0x00, 0x00
, 0xFC, 0xD6, 0xFC, 0xD6, 0xFC, 0x36, 0xF8, 0xEC, 0xF8, 0x1C, 0xE0, 0xF8, 0x00, 0xE0, 0x00, 0x00

  /*照準*/
, 0x00, 0xF3, 0x61, 0xF3, 0x41, 0xF3, 0x01, 0xE3, 0x01, 0x03, 0x00, 0x03, 0x00, 0xFC, 0x78, 0xFC
, 0x00, 0x9E, 0x0C, 0x9E, 0x04, 0x9E, 0x00, 0x8E, 0x00, 0x80, 0x00, 0x80, 0x00, 0x7E, 0x3C, 0x7E
, 0x00, 0xFC, 0x00, 0x03, 0x01, 0x03, 0x01, 0xE3, 0x41, 0xF3, 0x61, 0xF3, 0x00, 0xF3, 0x00, 0x00
, 0x00, 0x7E, 0x00, 0x80, 0x00, 0x80, 0x00, 0x8E, 0x04, 0x9E, 0x0C, 0x9E, 0x00, 0x9E, 0x00, 0x00

  /*敵*/
, 0x00, 0x07, 0x07, 0x18, 0x19, 0x26, 0x20, 0x5F, 0x20, 0x5F, 0x7E, 0xBF, 0x5E, 0xBF, 0x4E, 0xBF
, 0x00, 0xE0, 0xE0, 0x18, 0x58, 0xA4, 0xAC, 0x52, 0x54, 0xAA, 0x7E, 0xFD, 0x7E, 0xF9, 0x7E, 0xF1
, 0x40, 0xBF, 0x40, 0xBF, 0x40, 0xBF, 0x20, 0x5F, 0x20, 0x5F, 0x18, 0x27, 0x07, 0x18, 0x00, 0x07
, 0x16, 0xE9, 0x2E, 0xD1, 0x56, 0xA9, 0x2C, 0xD2, 0x5C, 0xA2, 0xB8, 0x44, 0xE0, 0x18, 0x00, 0xE0

  /*空白*/
, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

/*--------スプライトx個(縦×横)をまとめて表示*/
/* (スプライト番号 ,X座標 ,Y座標 ,キャラ番号) */
void sprite_set_locx(int iSpn ,int iLx ,int iLy ,int iCn)
{
        int iX,iY;

    iLx -= (MAS_PIX / 2);
    iLy -= (MAS_PIX / 2);

        for(iY=0 ;iY<MAS_PIX ;iY+=CHR_PIX)      {
            for(iX=0 ;iX<MAS_PIX ;iX+=CHR_PIX)   {
              sprite_set_char(iSpn,(CPN_MY-CPN_SP)<<CFSFT_PALETTE | iCn | CFM_SPR_UPPER);
              sprite_set_location(iSpn ,iLx+iX ,iLy+iY);        /*スプライト位置指定*/
              iCn++;
              iSpn++;
            }
        }
}
```


★影さんみたいな社会人になるにはどうすればいいんですか？（秋田県横手市・ハングンマル20歳）……【つぐ美：影さんは社会人のクズですから。編：そもそも，人間ではないですから。影：みんなでよってたかって！というわけで，うたれ強く，信念を持つこと。】

```
/*--------------------------スプライトを消す---*/
void sprite_clrx(int iSpn)
{
        int iCnt;
        for(iCnt=0 ;iCnt<MAS_CHR ;iCnt++) {
            sprite_set_location(iSpn ,224 ,144);          /*スプライト位置指定*/
        iSpn++;
        }
}


/*-------絶対値を求める関数----*/
/*引数 (比較したい値１ ,比較したい値２) */
int fncAbs(int d1,int d2)
{
    if(d1 > d2)
        return(d1-d2);
    else
        return(d2-d1);
}

/* ------------ 値0～X-1と循環させる関数 */
/*  引数(補正したい値 ,循環させたい範囲X) */
int fncLoop(int iArg ,int iPitch)
{
        if(iArg < 0)
                iArg = iArg + iPitch;

        if(iArg >= iPitch)
                iArg = iArg - iPitch;

        return(iArg);
}

/*-------------正負を求める関数----*/
/*引数 (比較したい値) */
int fncSgn(int d1)
{
    if(d1 > 0)
        return(1);
    else
        return(-1);
}

/* ------ 値が、1より小さいなら1とする関数 */
/*  引数(補正したい値 ) */
int fncFix(int iNum )
{
        if(iNum < 1)
                iNum = 1;

        return(iNum);
}

/*----------------敵情報構造体*/
typedef struct {
    int  x;    /*  */
    int  y;    /*  */
} ITI;

typedef struct {
    int  x;    /*  */
    int  y;    /*  */
    int  ang;  /* */
} KAK;
```

```
ITI itiTeki[TEKI_MAX];  /*敵の配列*/

/*-----------------------敵情報の初期化-*/
void teki_init()
{
    int iCnt;
    ITI *pIt;   /*敵配列用ポインタ*/
    pIt = itiTeki;
    for(iCnt=0 ;iCnt<TEKI_MAX ;iCnt++) {
        pIt->x = OBJ_NONE;
        pIt++;
    }
}



/*---------------------敵の発生---*/
void teki_bone()
{
    int iCnt;
    ITI *pIt;   /*敵配列用ポインタ*/

    pIt = itiTeki;
    for(iCnt=0 ;iCnt<TEKI_MAX ;iCnt++) {

        if(pIt->x == OBJ_NONE)
        {
            if(rand() & 1) {
                pIt->x = (rand() & 1)*224;
                pIt->y = (rand() % 144);
            } else {
                pIt->x = (rand() % 224);
                pIt->y = (rand() & 1)*144;
            }
            return;
        }
        pIt++;
    }
}

/*----------------------メインプログラム*/
void main(int argc, char *argv[]) {
#define KEYDELAY 2

        int iKey,iKeyb;   /*キー入力データ、キー確定カウンタ*/
    KAK kTama;      /*弾のＹ座標*/
        KAK kMy;        /*主人公座標*/
    int iBone;         /*敵の発生率*/
    int iSpeed;        /**/
    int iScore;        /*スコア*/
    int iTx,iTy;       /*テンポラリ*/
    int iCnt;          /*汎用カウンタ*/
    ITI *pIt;    /*敵配列用ポインタ*/

        kMy.x    =LCD_PIXEL_WIDTH /2;  /*主人公の座標*/
        kMy.y    =LCD_PIXEL_HEIGHT/2;
    kMy.ang =0;  /*主人公角度*/

        /*パレット設定(キャラクタパレット番号, LCDカラーパレット番号)*/
        palette_set_color(CPN_MY  , 0x0740);
        palette_set_color(CPN_TEXT, 0x7420);

        /* LCD階調データ初期化 */
        lcd_set_color(0x6420, 0xfca8);
          /* (LCDカラーパレット番号000-011のLCD階調,
               LCDカラーパレット番号100-111のLCD階調 ) */
```


★授業にて。先生「今日の私はよくしゃべりますねぇ。…副作用なんですよ。」……って何の？！（石川県金沢市・**村田春佳17歳**）……
【つぐ美：今日の影さんは静かですねぇ。**編：**なんか，不気味ですね。**影：**普段とちがった態度を見せ，周りの気を引くことも大事。**】**

```
		/* ディスプレイ設定 */
		display_control(DCM_SPR + DCM_SCR2 );	/* SPR/SCR2表示 */

	/*テキスト初期化*/
	screen_fill_char(SCREEN2 ,0 ,0 , 28, 18 ,(CPN_TEXT << CFSFT_PALETTE) | CHN_NONE);
	text_window_init(0 ,0 ,10 ,1 , CHN_TEXT);
	text_put_string(0,0,"SCORE00000");
	/* スプライト設定 (スプライト番号 ,スプライト個数) */
	sprite_set_range(SPN_ST, SP_MAX );

	/*キャラクタ設定(キャラ番号 ,キャラ数 ,キャラデータ) */
	font_set_colordata(CHN_START ,sizeof(cChrDat) / CHR_BYTE ,cChrDat);

	teki_init(); /*敵の初期化*/
	iScore=0;	/*スコアの0クリア*/
	iBone=100;	/*敵の発生度*/
	iSpeed=100;	/*敵のスピード*/
	kTama.x = OBJ_NONE;

		/*-------------メインループ*/
		for(;;) {
				sys_wait(1);	/*ウエイト*/

				/*-----------主人公のキー処理*/
			iKey = key_press_check();
		if(iKey == 0)
			iKeyb=0;

		if (iKey & KEY_START)
				break; /*プログラム終了*/

				/*-----キャラ移動 */

				if(iKey & KEY_RIGHT1) { /*右移動*/
			iKeyb = fncLoop(iKeyb+1 ,KEYDELAY);
			if(iKeyb==0)
				kMy.ang = fncLoop(kMy.ang+1 ,ANG_MAX);
		}

			if(iKey & KEY_LEFT1) {		/*左移動*/
			iKeyb = fncLoop(iKeyb+1 ,KEYDELAY);
			if(iKeyb==0)
				kMy.ang = fncLoop(kMy.ang-1 ,ANG_MAX);
		}

				if(iKey & KEY_UP1) {	/*移動*/
			iKeyb = fncLoop(iKeyb+1 ,KEYDELAY);
			if(iKeyb==0) {
				kMy.x += vec[kMy.ang].dx/50;
				kMy.y += vec[kMy.ang].dy/50;
			}
		}

				/*----主人公表示(スプライト番号 ,X座標 ,Y座標 ,キャラ番号) */
		sprite_set_locx(SPN_MY  ,kMy.x ,kMy.y ,CHN_MY);

		/*-----スコープ*/
		iTx = kMy.x + (vec[kMy.ang].dx / 4);
		iTy = kMy.y + (vec[kMy.ang].dy / 4);
		sprite_set_locx(SPN_SCOPE ,iTx ,iTy ,CHN_SCOPE);

		/*--------------弾*/
		if(kTama.x == OBJ_NONE) {
			sprite_clrx(SPN_TAMA);
/*				if(iKey & KEY_A) {  */
				kTama.x = iTx;
				kTama.y = iTy;
				kTama.ang = kMy.ang;
/*			}*/
		} else {
			kTama.x += (vec[kTama.ang].dx / 40); /*弾の移動*/
			kTama.y += (vec[kTama.ang].dy / 40);
				sprite_set_locx(SPN_TAMA  ,kTama.x ,kTama.y ,CHN_TAMA);

			if((kTama.x < 0)||(kTama.y < 0)||(kTama.x > 224)||(kTama.y > 144)) {
				kTama.x = OBJ_NONE;
			}
		}

		/*----------------------敵移動*/
		pIt = itiTeki;
		for(iCnt=0 ;iCnt<TEKI_MAX ;iCnt++) {
			if(pIt->x != OBJ_NONE) {
				if(rand() < (RAND_MAX / iSpeed)) {
					pIt->x += fncSgn(kMy.x - pIt->x)*2;	/* */
					pIt->y += fncSgn(kMy.y - pIt->y)*2;	/* */
				}
					sprite_set_locx(SPN_TEKI + MAS_CHR*iCnt ,pIt->x ,pIt->y ,CHN_TEKI);

				if((fncAbs(kMy.x ,pIt->x)<MAS_PIX)
				 &&(fncAbs(kMy.y ,pIt->y)<MAS_PIX)) {
					text_window_init(10 ,10 ,10 ,1 , CHN_TEXT2);
					text_put_string(0,0,"GAME OVER!");
					sys_wait(75);
					key_wait();
					return;		/* ゲームオーバー*/
				}

				if((fncAbs(kTama.x ,pIt->x)<MAS_PIX)
				 &&(fncAbs(kTama.y ,pIt->y)<MAS_PIX)) {
					pIt->x  = OBJ_NONE;	/*敵の消滅*/
					kTama.x = OBJ_NONE;	/*自弾の消滅*/
					iScore++;
					iBone  = fncFix(iBone-1);
					iSpeed = fncFix(iSpeed-1);
				}
			}else{
				sprite_clrx(SPN_TEKI + MAS_CHR*iCnt);
			}
			pIt++;
		}
		if(rand() < (RAND_MAX / iBone))
			teki_bone();  /*敵の発生*/

		text_put_numeric(5 ,0 ,5 ,NUM_PADZERO ,iScore); /*得点表示*/
		}
}
```

《リスト２》zerot2.cfのプログラム・リスト

```
name: zerot2
info: zerot2
mode: 7
source: zerot2.bin
output: zerot2.fx
```

《リスト３》makefileのプログラム・リスト

```
include ../makefile.inc
all:     zerot2.fx
zerot2.fx:   zerot2.bin
zerot2.bin:  zerot2.obj
zerot2.obj:  zerot2.c
```


★いまだに，MSXのMMLでMIDIをやっている私って，時代を逆流してる？（大阪府岸和田市・**紫猫20歳**）……【**影：**逆流というより，時代が進んでないだけじゃん。**編：**きついですね。**影：**たまには，きついことも言ってみる。**編：**前々ページから引きずってる？】

PC-E650用



対COM戦との撃ち合いが熱い

フォート　キャプチャー　じょうさいこうりゃく

Fort capture 城塞攻略

大沼　義憲（19歳）

CHECKER FLAG

つぐ美：真ん中にある壁を，いかに早く破壊するかがこのゲームのポイントですね。改造のアドバイスにもありますが，2人対戦プレイが面白そうな作品ね。

Dr.D：リスト面では，三角関数を使ってはおるが，ほとんどが基本的な命令で記述されておるので，内容の把握もしやすい作品と言える。COMの砲弾は乱数で決めておるようじゃが，ここにチョットした思考ルーチンを付け加えてみても面白いのう。

内容・使いかた・遊びかた

プレイヤーVS COMの砲弾対戦ゲームです。

SIN・COSを使用しているので，DEGREEコマンドを実行してください。RUNすると中央に壁，左端に自分の基地，右に敵の基地，そして自分の基地の上にメーターが表示されます。

どれかキーを押すとメーターが動き始めます。もう1度キーを押すとメーターのたまり具合に対応した高さの砲弾が発射されます。壁に当たると爆発し，その分の壁が破壊されます。それが終わると敵も同じように砲弾を発射してきます。それを繰り返し，弾が貫通して敵あるいは自分の基地に当たるとゲーム・オーバーです。このゲームはいかにして壁を壊していくかがポイントとなります。自分が壁に穴をあけたとたんにそこから敵の砲弾が飛んでくると結構悔しいです。

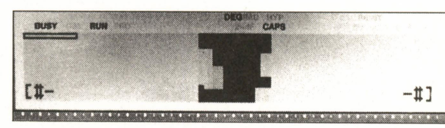
プログラムの仕組みと操作法

以前，ロープレもどきのものを投稿したのですが，200行と巨大すぎたので，今回はなるべく単純になるように努めました。すべてのキーが対応しています。押しやすい場所に決めてください。

改造アドバイスと苦労したところ

おおまかには自分ルーチン→敵ルーチンの連続です。人VS人の対戦ゲームに作り変えるのも簡単だと思います。

弾の変化には放物線の方程式を利用していますので，それなりにリアリティにも配慮しました。一応，先頭の行で砲弾のスピードを変えられるようにしてありますが，0.08以上の数値を入れると壁をすりぬける場合がありますので，ご注意を。



《写真》真ん中の壁をうまく破壊しよう

『Fort capture城塞攻略』プログラム・リスト

```
10 'ﾎｳﾀﾞﾝｽﾋﾟｰﾄﾞ
20 SPEED=0.06
30 ' ｶﾞﾒﾝｾｯﾃｲ
40 CLS
50 LINE (100,0)-(140,31),BF
60 LINE (0,2)-(30,0),B
70 LOCATE 0,3:PRINT "[#-"
80 LOCATE 36,3:PRINT "-#]"
90 ' ﾌﾟﾚｲﾔﾙｰﾁﾝ
100 WHILE (1)
110 QI=-1:QM=-1:T=0
120 IF INKEY$ <>"" THEN
130 RN=0:FOR W=1TO 200:NEXT W
140 '' ﾊｯｼｬﾀｲﾐﾝｸﾞ
150 WHILE (INKEY$ ="")
160 RN=RN+1
170 IF RN=30 THEN RN=0:LINE (1,1)-(29,1),R
180 PSET (RN,1)
190 WEND
200 LINE (1,1)-(29,1),R
210 XK=COS RN:YK=SIN RN
220 IF XK<0 THEN XK=-XK
230 IF YK<0 THEN YK=-YK
240 G=14:L=10:V=50
250 FOR T=0TO L STEP SPEED
260 M=-V*T*YK+(G*(T^2)/2)+31
270 I=V*T*XK*(3/2)+15
280 PRESET (QI,QM):GOSUB 490
290 IF (I>=220)*(M>=25) THEN GOSUB 500:LOCATE 0,1:PRINT "WIN!":END
300 QI=I:QM=M
310 NEXT T
320 ENDIF
330 ' ﾃｷﾙｰﾁﾝ
340 IF T>=10 THEN
350 RN=RND 40
360 XK=COS RN:YK=SIN RN
370 IF XK<0 THEN XK=-XK
380 IF YK<0 THEN YK=-YK
390 FOR T2=0TO 10 STEP SPEED
400 I=-V*T2*XK*(3/2)+200
410 M=-V*T2*YK+(G*(T2^2)/2)+31
420 PRESET (QI,QM):GOSUB 490
430 IF (I<=15)*(M>=23) THEN GOSUB 500:LOCATE 35,1:PRINT "WIN!":END
440 QM=M:QI=I
450 NEXT T2
460 ENDIF
470 WEND
480 'ﾁｬｸﾀﾞﾝ
490 IF POINT (I,M) THEN
500 X2=1:FOR X1=1TO 5STEP 1
510 LINE (I+X1,M+X1)-(I-X1,M-X1),B:LINE (I+X1,M+X1)-(I-X1,M-X1),R,BF:NEXT X1
520 T=L:T2=10
530 ELSE
540 PSET (I,M)
550 ENDIF
560 RETURN
```


★《今月のベーマガお楽しみプレゼント》①『海洋科学技術センター取材時ゲット商品』…海の深層水，ボールペンなど／1名様（提供：編集部）　②『AMDロゴ入りボールペンと卓上オートカレンダー電卓』／1名様（提供：日本AMD）　③『キャッチャー商品』…なにが



CMM試聴室

毎月みなさんの曲を紹介するコーナーです。募集要項をご覧のうえ，どしどしご応募ください。ここで紹介しているデータは，CMMのWebSite（http://www.basicmagazine.net/cmm/）に収録されています。曲への意見，アドバイスなども大歓迎します。

南米の風 ……自作

投稿者：さば夫さん（23歳）

E-Mail：sabao@spa.att.ne.jp

制作音源 SC-88ST Pro

ストリートライブの音楽に影響を受けました

久しぶりの投稿になります。かなり以前に，CMMに投稿していた者です。

笑点とか，送っていましたね(笑)。

94年ぐらいから2，3年ほどずっと購読していましたが，ふっと店頭に並ばなくなったなと思ったら，ベーマガと合体していたのですね。

曲についてですが，札幌大通公園周辺でストリートライブをやっている南米の民族音楽を演奏するバンドの影響を受けて作ったものです。しかし，どこかしらテレビの大自然系ドキュメンタリー番組のテーマ曲みたいな雰囲気がまざってきました。

楽器編成はそのバンドとほぼ同じような，エレキベース，ギター，パーカス，現地の笛系の楽器（オカリナで代用）それにストリングスとブラスセクションを加えてやや壮大にしました。ドキュメンタリー風になったのはそのせいですね。音色やエフェクトとかはあまりいじらず，ナチュラルな感じにしてます。

1年ほど前に作曲したものですが，なかなか気に入っているので投稿することにしました。曲名を考えるのが苦手なので，最初は「南米風の曲」と言っていました。それじゃああんまりだろうということで，今回，「南米の風」と，ややそのまんまなタイトルをつけました。

現在，紆余曲折を経てゲーセンに設置されるスロットの効果音を作ったりさせてもらっています。これもかつてのCMM投稿の日々があったおかげかもしれません。感謝しております。

編集部から

ペンネームは変わっていますがなつかしい方からの投稿です。「笑点」はイントロの数小節を聴いた瞬間に採用を決めました。当時とはだいぶ様変わりしていますが（今月は1ページですし），これも時の流れというものでしょうか。

昔話はさておき，曲ですが，ミックスの全体的なまとまりにはやや欠けますが，曲の構成などはとてもよくまとまっていると思います。ミックスについては，同系統の楽器（ラテンパーカッションなど）がある程度まとまって聴こえるようにすることと，中央に芯の通ったパートがあると聴きやすいかなと思いました。後半にどんどん場面が切り替わる部分が好きです。

Face to Face ……自作

投稿者：HKRさん（15歳）

E-Mail：webmaster@hkr.port5.com

URL:http://hkr.post5.com/index.chtml

制作音源 SC-88Pro

ダンス系ドラムの作成に力を入れました

どうも。今回もかなり久々の登場となってしまっているHKRです。

今回の曲は基本的に「ダンス系ドラムは自分の能力でどこまで作れるか」を主観に置いたものの，最近のダンスドラムをちゃんと知らない人間なので結局中途半端に終わってしまっています。後々考えてみると，単に重いバスドラつけてハンドクラップまがいのスネアをつけているだけのような気もしますが。

曲の構成はというとまったく放りっ放しで全然考えていません。あえて言えば音色選びを念入りにしただけです。全然駄目駄目です。

ドラムが全体的に三連符の連続になっていて，ノリがいい…はず。あっ，ちなみに最大同時発音数32音の88ProでもOK。この曲のドラムが良いと思ったら，ぜひ初期化部分とか色々コピーして使ってあげてください。

編集部から

ハウス系のリズムを持つ明るい曲です。ユーロビート的な明るさもちょっと混じっているでしょうか。こちらもミックスのまとまりをもうちょっとがんばってほしいですが，曲の持つ雰囲気は聴きやすいしとてもいいものだと思います。ドラムに関しては，とくにバスドラムがしっかり聴こえるのはいいですね。SC-88系の音源は発音数が重なるとバスドラムのディケイが途切れてしまうことがあり，NRPNでディケイを伸ばすと解決する場合があります。スネアの連打部分でかなりテンポがよれてしまっているので，他のパートのアタックを前後に微妙にずらすと改善されると思います。

マイコンBASICマガジン

明日のスター☆プログラマー

今月のベスト・プログラマー

つぐ美：今月はどの作品がベストにふさわしいか，さっそくみんなの意見を聞いてみましょう。

影：わたしは，F-BASIC用の『マウスde世界陸上』がお気に入りだ。短い時間で手軽に遊べるのがいいね。ランキング機能もあるので，ついやり込んでしまう。

編：『マウスde世界陸上』は，確かにすばらしい作品ですね。同じく超大作作品では，VB用の『Logicer』もいいですよ。問題作成までは，チョット手間がかかるのですが，画像ファイルからロジックの問題を作成できるというのは，良いアイデアです。超大作以外の作品も，しっかり作られている作品が多いのですが，どの作品もアイデアがスタンダードなものなので，新鮮味には欠ける印象でしたね。

Dr.D：ワシも影クンと同じく，『マウスde世界陸上』が目についた。マウスという，ゲームには不向きのはずのデバイスで，ここまで遊べるゲームは市販されているものを含めても珍しいのではないかな？完成度の高い作品じゃ。

つぐ美：私も今月は『マウスde世界陸上』がイチオシ。チョットプレイするつもりでも，ついハマってしまうんですよね。

編：それでは，今月は超大作の作品ではありますが，『マウスde世界陸上』がベスト・プログラムにふさわしいみたいですね。みなさんよろしいですか？

一同：異議な〜し。

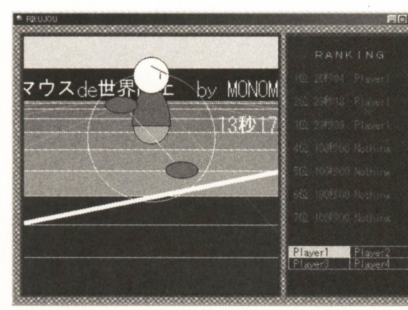
編：というわけで，今月号のベスト・プログラマーは，『マウスde世界陸上』の作者MONOMONさんに決定です。ちなみにMONOMONさんは，2000年のスター・プログラマーでもあります。2年連続受賞の可能性もでてきましたね。

影：ところでDr.，今月からベスト・プログラムを使って，新たな解説ページを始めるという話を聞いたんですが？

Dr.D：うむ，くわしくはそのページでも触れるが，せっかくのベスト・プログラムなのじゃから，その良さを読者にもっと知ってもらおうというわけじゃ。

つぐ美：普段からCHECKER FLAGでいろんな作品の評価をしているDr.Dの解説ですから期待しちゃいますね。

編：これからは，この「明日のスタープログラマー」，「Programmer's Room」と合わせて，新企画ページ「Dr.Dのベスト・プログラム研究室」の応援もよろしくお願いします。



▲F-BASIC用の『マウスde世界陸上』

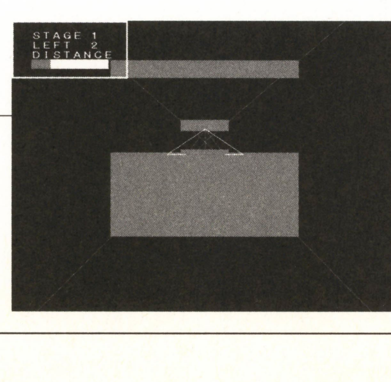
今月の掲載者から一言

Windows（Visual C++ V6.0）用
「Walls in the Box」
夏夕樹（19歳）

今現在，ベーマガでは，DirectXというゲーム作りにはかかせないものを使ったプログラムは，初期化なども含めてしまっているので，超大作プログラムにせざるを得ないという状況にあります。これは本格的なゲーム・プログラムを，雑誌に掲載できないことはもちろん，C言語でのゲーム・プログラム投稿の困難さに結びついてしまっています。

今回，私はライブラリを使ったプログラム部分のみを投稿プログラムとする形をとったのは，上記のような状況を打破，およびライブラリを使ったプログラムを組むことによりC言語でもゲーム・プログラムが投稿できるということをみなさんに広く知ってもらうためです。

今のところライブラリを使ったプログラム投稿は一つも掲載されていないようですが，「ゲーム・プログラムの本質」を語るにはDirectXの初期化や使いかたなどは考慮する必要はないと，つまりライブラリを使ったプログラムでも投稿するに値するプログラムだと思っています。



投稿ありがとう

富野裕太／峯本祐麻／東口克己／宇品忠司／石山明弘／R-SOFT／tayosan／堀内伸恭／南とらいば／マーサン／tao／荒木伸夫／MONOMON／たくちゃん／usk／@SYD／日下心平／YUITA／NAK／Shota／遊牧家族／大山順彦／金井義男／山路桃香／大沼義憲／夏夕樹／タカヒ／寺坂龍太／Mr.親父／どんと／宮迫　理／後藤道彦／間瀬秀聡／KIKUCHIくん／今野脩一郎／小山　勉／森次／金井真彦／十文字直樹／NAOKI／さば夫／そめっと／ナガタヒロカズ
※全員紹介できなくてごめんなさい

投稿プログラム・コーナー
Programmer's Room

今月の総評

編：8月号は1年間の中でも，投稿数が多い月の一つだったりします。そんなわけで，今回採用に至らなかった作品でも，普段なら採用レベルの作品もありました。そのようなわけで今月は採用されなかったが，レベル的に高いものは保留にしてあります。というのは，次回チェック分の作品しだいでは，保留作品から採用される場合もあるということです。

でも，ここで，“一口”や“もう一歩でした”として紹介した作品は残念ながら保留作品には入っていませんので，ぜひとも，このアドバイスを参考に改良を加えるなどして再度投稿してもらいたいと思います。

また，最近は超大作の投稿が多く，通常よりも1～2か月遅れて掲載されたりします。超大作を投稿したのに名前すら出ていないゾ，という方は掲載まで気長に待っていただければありがたいです。

期待のiアプリ投稿，投稿数もだんだんと増えてきています。ですが，エミュレータのみで動作確認をした，という作品が多く，実機でテストしてみると不具合が出てくる作品もあります。iアプリが動作する携帯電話は，まだまだ値段がはるので持っていない読者が多いのだとは思いますが，どうにかして一度は実機でのチェックをお願いします。

また，採用されなかった作品に多いのは，「チェック不足から生じる不備」という内容。このチェック不足というのは，動作不具合やバグはもちろん，ゲーム作品で言えば難易度調整などにも当てはまります。作成した作品は機会があれば，他の人に一度見てもらうのがいいですよ。人に見せる機会がない，という人は期間をおいてチェックし直すという手があります。作ったらすぐに投稿したい，という気持ちはわかりますが，そこをグッと我慢して動作チェックに時間をかけてみてはいかがでしょうか？

一口 CHECKER FLAG

●iアプリ用『bar and ball』／Mr. 親父（18歳）

編：参考にしたサンプルに，得点を追加しただけではもの足りません。また，実機での動作に不具合があります。

●Visual Basic 用『三つの駒』／R-SOFT （17歳）

つぐ美：ルーレット主体の内容なので，ゲーム結果にプレイヤーの思考がほとんど関係せず，おもしろ味に欠ける内容です。

●C 言語用『記憶力マスター』／堀内伸恭（28歳）

影：ホントに基本的な内容なので，こちらもおもしろ味に欠ける。ゲーム面でのアイデアもしっかり考えてほしい。

●HSP用『自然な継ぎ目エディタ』／南とらいば（24歳）

つぐ美：画像ファイルを加工する内容なので，実用性で中途半端な面があるかも？画像ファイルを作れる人は，この作品でできることを自分でやってしまうだろうし…。

●VisualBasic用『RADAR～ネッシーと悪徳館長と生物学者～』／宇品忠司(24歳)

Dr.D：全体的な出来は良いのじゃが，ルールが理解しにくく，何回も遊ぶ気になれない。プレイヤーにわかりやすい設定にするという工夫も必要じゃ。

●Excel（VBA）用『お宝捜し』／山路桃香（？歳）

編：マインスイーパ風のゲーム。画面に現れる数字のヒントと，宝の位置が一致していないバグがあります。テストプレイはしっかり！

●VisualBasic用『RANDOM CODE』／YUITA（17歳）

影：文字をランダムに暗号化するというアイデアはGood。だけど，10Kバイトのファイルまで，という制限が使い勝手を悪くしている。この部分の修正を期待。

●N88-BASIC用『春山波』／大山順彦（？歳）

Dr.D：グラフィック命令を使った簡単なプログラム。こういった単純な命令で画像を作るという考えは良いのじゃが，もの足りないという印象を持つのも事実。アニメーション効果など，演出面での工夫もしてみてはどうかな？

●VisualC++用『Fire Crash Ver.1.0』／ピュー太郎（32歳）

編：画面に存在する炎を動かして敵を倒す，というシステムは，独特ですが遊びにくいという印象があります。このシステムを生かす工夫がほしいところ。

●VBScript+DirectX用『Sound Mix』／金井義男（39歳）

影：基本的には，以前に本誌紹介で紹介した二つのサンプルを組み合わせた内容の作品。うまく組み合わせたプログラム面での技術は評価できるけど，やはりオリジナリティのある作品が見たいなぁ。

●Macintosh(FutureBasic)用『めてぱに』／tao（27歳）

編：ベーマガでは珍しいMacのプログラム作品の投稿。基本的なゲームで，初心者には参考になる内容だとは思うのですが，ゲームとして見るともうチョット改良点があるかな，という内容。レーザーを撃つたびに敵の動きが止まるのは，ゲームの流れを壊していますね。

今月はもう一歩でした

●HSP用『スピンパネル』／遊牧家族（18歳）

つぐ美：先月掲載したHSP用の『16パズル』と同じようなアイデアの作品。こちらの作品は，パネル自体を回転させつつ移動させる，というもの。アイデア面では良かったのですが，前述の作品があっただけに，決め手に欠けてしまったという印象です。演出面での改良がほしいところかしら？

Dr.Dの LABORATORY ベスト・プログラム研究室



今月のベスト・プログラム：F-BASIC用『マウスde世界陸上』（66ページ掲載）

Dr.Jに負けじとばかり，今月から新たに立ち上げられたのがこのコーナー，その名も「Dr.Dのベスト・プログラム研究室」。以前から要望の強かったリストのアルゴリズムや各処理方法の解説を毎月ベスト・プログラムに選ばれた作品を対象に行おうという試みじゃ。第1回目の対象プログラムはF-BASIC用の『マウスde世界陸上』。ゲーム内容は単純ながらもスタート前の拡大表示やマウスによる操作など特徴的な処理が多く使用されておる作品じゃ。今回はこの「拡大処理」と「マウスによる移動処理」をピックアップして解説していこう。 **（Dr.D）**

拡大処理

この拡大処理は主に**リスト1**の部分で行われておる。ここで注目してほしいのは“Z”という変数があることじゃ。本来2D画面の描画ではZ座標は必要ないが，ここではいわば擬似的にZ座標を用意し，Zの値（正確にはこのプログラムでは，この値は倍率を示し，20のときにカメラが最も接近している場所を示す。ちなみにスタート時のZの値は1であり，このときカメラは遠い場所にあることを示す）を利用して，拡大率を考慮したX,Y座標の値を求めておるのじゃ。

このような処理は擬似的な3D処理を行ううえでの基本。過去のベーマガのGet &Runのコーナーでも似た形式のサンプルを紹介したことがあるので，覚えている読者もいるのではないかな？こういった奥行きを表現する処理を取り入れると，見栄えのする作品になるのでぜひ挑戦してほしい。

マウスによる移動処理

主に**リスト2**，**リスト3**の部分でマウスによる移動を実現しておる。**リスト2**には作者が記述したコメントもあるので，各行でどのような処理を行っているかが，わかるじゃろう。その中で「総移動距離」と「現在の半径」は三平方の定理を使って，現在と以前のマウスのX,Y座標を結んだ直線を求めておる。

リスト2で得た値を用いて，速度ならびにキャラクターの位置を求めているのが**リスト3**の記述となる。

ちなみに，今回の処理方法では，マウスを左回しで動かすと，逆方向に走り出すという現象が起きる。これは，バグというわけでなく，むしろマウスの動きから正確にキャラクターの動きを再現したからこそ，起きる現象と言えるじゃろう。

改造してみよう

ここまで解説してきた二つの処理を，より深く理解するために，プログラム改造のヒントを記述していこう。

まずは「拡大処理」から。この処理はZの値を増やすことで，拡大を実現しておる。では，反対に画面から遠ざかっていくような処理はどうすればできるじゃろうか？

逆に考えればすぐにわかりそうなことじゃが，要はZの値を小さくしていけば，反対の動作が可能になるはずじゃ。これは，**リスト1**の中の解説コメントにも記述してあるので，参考にしてほしい。

注意したいのは，この遠ざかる処理を作成する場合は，Zにかかわる式を変更するだけでなく，新たな終了条件（どこまで遠ざけるかを記述した式）を持った制御文（for文やwhile文）の中で処理を行う，という点じゃな。

つぎは，「マウスによる移動処理」。このマウスによる処理は非常にうまく組まれており，簡単にプログラム改造できるルーチンではなさそうじゃ。

そこで，簡単に改造でき，かつ，はっきりと改造の結果がわかる**リスト3**のランナーの速度計算式に注目してみよう。ランナーの速度を変えるには，**リスト3**の1行目の記述を工夫すれば可能じゃ。

しかし，この式も見た目がチョット複雑で改造しにくい。そこで，注目したいのが，2行目のX座標を計算している式。ここでX座標の値を多く加算するように記述すれば，速度の原理部分などを理解しなくても，簡単に速度改造が可能じゃ。元の式をいじりたくない，という人は**リスト3**にも記してあるように，2行目の式の下に「X=X+SP」（SPは自分で定義したX座標に加算する値）といった式を加える方法もあるゾ。

最後に

ベスト・プログラムの解説いかがじゃったかな？はじめての試みなので，ワシもまだまだ不慣れな部分があるのじゃが，これから毎号，ベストに選ばれた作品をこのような感じで解説していこうと思っておる。そこで，ぜひともこのコーナーに対する意見や，感想などを聞かせてほしい。

アンケートハガキへの記述や，プログラム・コーナー宛てのハガキの他，お手軽なeメールによる投稿（送り先は，webmas@basicmagazine.net 件名は「Dr.Dのベスプロ研究室」など）や投稿プログラムの応募カードの裏などに記述してくれてもOKじゃ。みんなの意見を待っておるゾ。



解説するリストの抜粋

《リスト1》 拡大処理のリスト

```
while Z<20: gosub *TIMER_COUNT: Z=Z+DT/(300/Z): if Z>20 then Z=20:'カメラ接近  ※TIMER_COUNTはウェイト処理(下記参照)
  X=X0*((20-Z)/19)/Z: if GF>RUN_THR then gosub *FIELD else gosub *TRACK   ※FIELD, TRACKはそれぞれ画面
  DRAWBITMAP BUF,16,16: wend                                               を描画するサブ・ルーチン

TIMER_COUNTルーチンのリスト
*TIMER_COUNT :'======================================タイマーカウント========
A=GT: while GT<A+WT: GT=GTC&: GT=GT-(GT<A)*2^32: wend: DT=GT-A: return
※GTC&の定義：declare function GTC& lib "kernel32" alias "GetTickCount"()
```

解説コメント

変数Zの値：スタート時はZ=1であり，競技実行前のカメラの位置を示す。Z=20のときは競技実行時のカメラの場所になる。このルーチンでは，Zを徐々に増加させることで，カメラのズームを実現している。

式X=X0 *((20-Z)/19)/Z：Zから求める画面のX座標の計算式。X0には，このルーチン実行前にXの値が代入されている。Z=1のときは，X=X0 *((20-19)/19)/1→X=X0 * 1となり，X座標はX0そのままの値になる。

拡大途中のZ=15の場合は，X=X0 *((20-15)/15)/1→X=X0 * 1/3になりXはX0の1/3の値を示す。

《リスト2》 マウス処理のリスト

```
*MOUSE_COUNT :'====================================マウス操作カウント========
gosub *TIMER_COUNT                       :'ﾀｲﾏｰｶｳﾝﾄ
MXO=MX: MYO=MY: DO=D: RO=R               :'古い情報を記録
MX=mouse(0): MY=mouse(1)                 :'現在の座標
N=N+1                                    :'要素数
MV=MV+sqr((MX-MXO)^2+(MY-MYO)^2)         :'総移動距離
R=sqr((CX-MX)^2+(CY-MY)^2)               :'現在の半径
TR=TR+R                                  :'半径の累積
D=fnDIR(MX,MY)                           :'現在の角度  ※fnDIR：def fnDIR(X,Y)=atn((Y-CY-.1)/(X-CX-.1))/PI2-.5*(X<=CX)
DRT=D-DO-clng(D-DO)                      :'今回の回転角度
ROT=ROT+DRT                              :'総回転数
```

《リスト3》 速度ならびに座標計算式のリスト

```
V=V-DRT*R/1000-V*DT/(1000+R*10)   :'速度(回転で加速、時間で減速)
X=X+V*DT/50                       :'座標
※ここに「X=X+SP」といった式を記述することで，スピードの調整が可能(SPは自分で定義した変数とする)。
```

解説コメント

式X=X+V * DT/50：DTはリスト1に記述してあるTIMER_COUNTで得た値。1行目で求めたVに，DT/50の値を掛けた値がX座標に加算されていき，キャラクターが移動していく。

PlayStation 2でゲームが作れる！



「BASIC STUDIO パワフルゲーム工房」入門

第4回 背景を表示してみる

文／山崎　寛

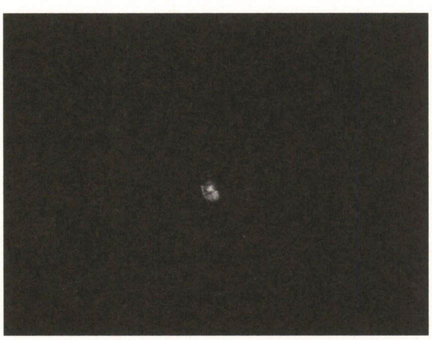
先月号では，スプライトでキャラクターを表示しましたが，今回はこれを応用して背景のマップを表示してみることにしましょう。先月の内容を思いだしつつ楽しんでくださいね。

『BASIC STUDIO パワフルゲーム工房』
発　売：アートディンク
対応機種：PlayStation 2
メディア：CD-ROM 2枚組
発売中
価　格：9,800円（税抜）
http://www.artdink.co.jp/japanese/title/bs/index.html



背景もスプライトだ

先月号では，RPG風のキャラクターをパッドで動かせるプログラムを作ったのですが，キャラクターの後ろは真っ黒で何もなく，ちょっと寂しい画面でしたね（**写真1**）。



《写真1》先月号のサンプル実行画面。背景がないのでちょっと寂しい

そこで，今月はパッと明るい背景を表示してみることにしましょう。

背景を表示するには，キャラクターを表示したのと同じくスプライトを用います。

他機種でプログラムの経験がある人は「背景はグラフィック画面やBG画面じゃないの？」と疑問に思うかもしれませんが，EZ-BASICには従来のグラフィック画面やBG画面などは存在しないので，スプライトを使うしか方法がないのです。

スプライトで埋め尽くすには

RPGの背景画像では，一つの画像の大きさが画面上では32×16ドットとなるので，画面全部を覆うようにするには，全部で280個のスプライトを使うことになります。何番を使ってもいいのですが，今回は100～379番のスプライトを使うことにします（**第1図**）。

280個もスプライトを使って平気なのかと思う方もいるかもしれませんが，EZ-BASICでは全部で1000個のスプライトを扱うことができます。なので，とりあえず数量としてはまだ大丈夫です。

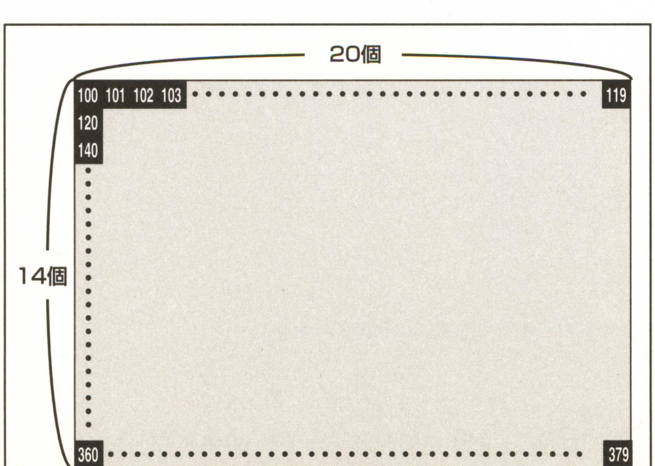
背景として表示するテクスチャは，EZ-BASICテクスチャ9ページ目の右上にあるRPG用の地形画像6種類です。便宜上，左上のほうから0～5の番号をつけ，この番号で背景の種類を区別することにします（**第2図**）。

実際に画面を埋め尽くして表示するようにしたサンプルが**リスト1**です。横20個×縦14個のスプライトを表示するため，FOR文で二重のループを作っています。

まず最初に変数Rに背景の種類を乱数で決めた値を入れています。この数値は0～5の6種類です。

変数Kがそれぞれのスプライトの番号で，左上が100で順番に増えていき，379まで使っています。

つぎにスプライトのパレット設定，表



《第1図》画面を埋め尽くすには280個のスプライトが必要だ

《リスト1》RPG用背景で画面を埋め尽くすサンプル

```
100 !初期設定
110 INIT
120 RANDOMIZE
130 !背景を描く
140 FOR J=0 TO 13
150   FOR I=0 TO 19
160     R=INT(RND*6) ―――― 地形の種類をランダムに決定
170     K=J*20+I+100 ―――― スプライトの番号を求める
180     SPRPAL K,11,0,1        ┐
190     SPRDISP K              ├ スプライトの表示設定
200     SPRPOS K,I*32,J*16,0   ┘
210     TX=MOD(R,4)*32+384 ┐
220     TY=INT(R/4)*32     ┘ テクスチャの位置を求める
230     SPRTEX K,9,TX,TY,TX+32,TY+32 ―― テクスチャを設定
240   NEXT I
250 NEXT J
```

示，表示位置設定を行っています。

TX，TYという変数は背景の種類によって決まるテクスチャの位置を求めるために使っています。乱数で決まる変数Rの値と対応するテクスチャの位置は**第1表**のようになっていて，これを210行〜220行で求めています。

実行すると**写真2**のように，画面いっぱいに適当な地形を表示して終了します。

配列変数を使おう

続いて先月号の内容を応用して，表示した地形の上をパッドで歩けるようにするのですが，その前に少し説明しておくことがあります。

これまでテキスト画面でゲームを作っていた際，画面上に描かれた文字を直接読みとって衝突したかどうかなどの判断をしていました。

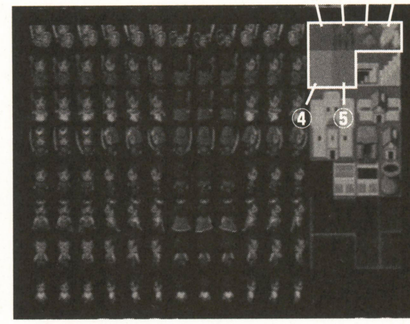
ところが，今回スプライトで描いた背景は画面から読み取ることができません。このままでは，キャラクターがどんな地形の上にいるかをプログラムから知ることができないのです。

《第1表》背景の種類と対応するテクスチャの位置

種類	位置
0	(384,0)
1	(416,0)
2	(448,0)
3	(480,0)
4	(384,32)
5	(416,32)

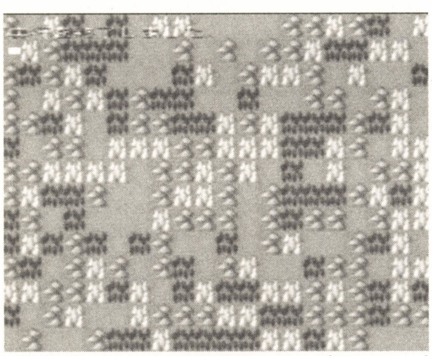
そこで，画面と同じ大きさの2次元配列を用意して，背景を描くと同時にその種類を配列変数に入れておきます。そうすれば，どこにどんな地形があるかをいつでもプログラムの中で知ることができます（**第3図**）。

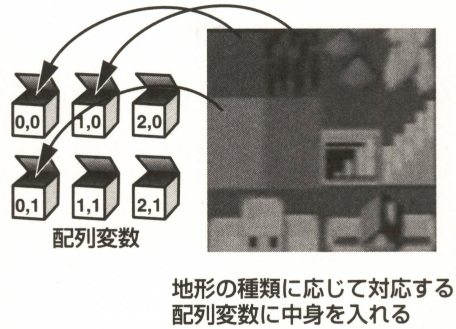
リスト2は，背景を描き，パッドでキャラクターを動かすというサンプル・プログラムです。基本的に**リスト1**と先月のサンプルを組み合わせた形なので，それらと比較するとわかりやすいと思います。



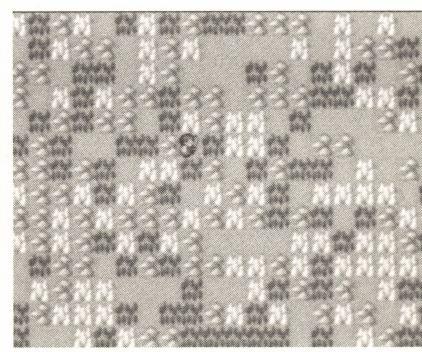
《第2図》右上の6種類のテクスチャを背景として使う

パッドで上下左右に動かせますが，地形の中で1種類だけ上を通れないようにしてあります。MAP()という配列変数でその処理をしているのですが，そのへんもどうやっているかうまくプログラムを読んで参考にしてください（**写真3**）。


《**写真2**》リスト1の実行画面。ランダムにマップを作って表示する



《第3図》配列変数を使えば，マップの内容がわかる


《**写真3**》パッドでマップ上を動かすサンプル・プログラム

《リスト2》マップ上でキャラクターを動かすサンプルプログラム

```
100 !初期設定
110 INIT
120 DIM P(4), MAP(0 TO 19,0 TO 13)
130 P(1)=0
140 P(2)=32
150 P(3)=0
160 P(4)=64
170 X=320
180 Y=112
190 PO=1
200 MU=0
210 !スプライト設定
220 SPRPAL 1,11,0,1
230 SPRDISP 1
240 !背景を描く
250 FOR J=0 TO 13
260   FOR I=0 TO 19
270     R=INT(RND*6)
280     MAP(I,J)=R
290     K=J*20+I+100
300     SPRPAL K,11,0,1
310     SPRDISP K
320     SPRPOS K,I*32,J*16,0
330     TX=MOD(R,4)*32+384
340     TY=INT(R/4)*32
350     SPRTEX K,9,TX,TY,TX+32,TY+32
360   NEXT I
370 NEXT J
380 !メインループ
390 DO
400   TX=X
410   TY=Y
420   IF PAD(1,4)>0 THEN
430     MU=0
440     PO=MOD(PO,4)+1
450     Y=Y+2
460   END IF
470   IF PAD(1,2)>0 THEN
480     MU=96
490     PO=MOD(PO,4)+1
500     X=X-4
510   END IF
520   IF PAD(1,3)>0 THEN
530     MU=192
540     PO=MOD(PO,4)+1
550     Y=Y-2
560   END IF
570   IF PAD(1,1)>0 THEN
580     MU=288
590     PO=MOD(PO,4)+1
600     X=X+4
610   END IF
620   FOR I=0 TO 1
630     FOR J=0 TO 1
640       IF X<0 OR X>608 OR Y<0 OR Y>208 OR MAP(INT((X+I*31)/32),INT((Y+J*15)/16))=3 THEN
650         X=TX
660         Y=TY
670       END IF
680     NEXT J
690   NEXT I
700   SPRTEX 1,9,MU+P(PO),0,MU+P(PO)+32,32
710   SPRPOS 1,X,Y,1
720   FSYNC 3
730 LOOP
```

（リスト中の注記）
- P(4)：キャラクターの動きのポーズ用
- MAP(0 TO 19,0 TO 13)：地形マップを入れる
- 130〜160：動きのポーズ用
- 170〜180：キャラクターの位置
- 190：ポーズ用
- 200：向き
- 210〜220：キャラクタのスプライト
- 280：マップ用配列に地形の種類を入れる
- 240〜370：背景を描く
- 400〜410：現在のキャラクター位置を保存
- 420〜610：キャラクターをパット操作に応じて移動
- 620〜690：キャラクターが画面から出ようとしたり通れない地形の上に来たら、元の位置にもどす

ワンダースワンのソフトを自作しよう!

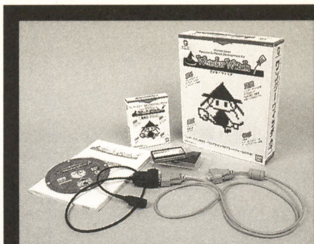
チャレンジ!! ワンダーウィッチ

ワンダースワン ホビープログラミングツール Wonder Witch ワンダーウィッチ



Challenge WonderWitch

第10回「インターネット接続に挑戦」



ワンダースワン・ホビープログラミングツール
『ワンダーウィッチ（WonderWitch）』
定価16,800円（税別）

◆パッケージ内容
・PC-WonderSwan 接続ケーブル
（スワンケーブル，クロスケーブル）
・ワンダーウィッチ専用カートリッジ
・CD-ROM　・マニュアル

◆動作環境
機種: PC/AT 互換機
OS: Windows 98／Me／NT4.0／2000
CPU: Pentium 133MHz以上
メモリ: 32MB以上（推奨64MB以上）
ディスプレイ: 640×480，256色以上
シリアルポート: 9600/38400bps
＊専用カートリッジのみ単体の購入も可能。
3,980円（税別）。

◆製品について問い合わせ
wonderwitch-info@bandai.co.jp

◆通信販売について問い合わせ
netshop@bandai.co.jp

今月は「WonderGate 対応プログラム開発キット β版」についてです。
（松原拓也）

今をさること1年前の2000年7月にバンダイとNTT ドコモから「ワンダーゲート（モバイルワンダーゲート）」が発売されました。これは携帯ゲーム機でインターネット接続を実現する画期的な商品だったのですが，競合する i モードに押されてか需要は伸びず，半年後にはヨドバシカメラで980円となって市場から消えていきました……かに思えましたが，このたび「WonderWitch 用 WonderGate 対応プログラム開発キット β版」が公開されました。

これはワンダーウィッチでワンダーゲートを使ったプログラミングを可能にするライブラリです。遂にワンダーゲートにまで対応させてしまうとは，キュート恐るべしです。

「ワンダーゲート」
バンダイ／NTTドコモ（「モバイルワンダーゲート」として販売）
実売価格 980～2979円

◆WonderSwanWeb
http://www.swan.channel.or.jp/
◆NTTドコモ「Mobile WonderGate」のページ
http://www.nttdocomo.co.jp/mobile/lineup/wonder/index.html

■携帯電話接続アダプタ部
PDCインターフェース部：デジタル携帯・自動車電話回線，9600bps(NTTドコモ)
電源：単4アルカリ乾電池1本，電池寿命約3時間（連続接続時
外観仕様：外形寸法70.0×43.0×25.0(mm)，質量50g（乾電池含む），ケーブル長13cm
■mopera接続用カセット部
メモリユーザーエリア約128Kバイト，表示機能 全角18文字×10行（12ドット時）
フォント10，12，14ドット（標準12ドット）
使用文字種JIS第1・第2水準漢字，非漢字
入力方法 ソフトウエアキーボード入力
■アプリケーション
・ブラウザ機能 HTML3.2相当に対応
表示可能画像 GIF
ブックマーク登録 20件
・メール機能 対応プロトコル SMTP/POP3
アドレス帳 20件
保存メール 受信ボックス/128件（32Kバイト）まで
送信ボックス/32件（16Kバイト）まで
・ダウンロードゲーム 保存エリア 32Kバイト（ゲームは1つまで保存可能）

購入方法

ほうぼうを歩き回って「ワンダーゲート」を探してみましたが，残念ながら店頭からかなり消え去っています。

結局，東京・渋谷の「さくらや渋谷店」で980円で購入しました（5階のゲーム関連でなく，1階の携帯端末のフロア）。なお，さくらや新宿店や，「さくらやインターネットショップ」（http://www.sakuraya.co.jp/）でも980円で購入できます（6月現在）。ネットの通販は会員登録が必要で，さらに送料が加わります。

単体で使ってみる

まず，付属の「mopera」対応カートリッジでワンダーゲートを使ってみました。ワンダースワンの周辺機器差し込み口に，電池を装てんした携帯電話接続アダプタを取り付け，接続ケーブルをデジタル携帯電話に接続します（写真1）。最初は「mopera」というサービスに設定されていて，プロバイダの登録なしでいきなり接続できます。

これでインターネットを介した正真正銘のWEBブラウジングとメールが楽しめるのは驚きですが，速度が遅く操作性が悪いです（画面1）。



▲《写真1》乾電池まで付属する徹底ぶりで，買って即ネット接続

ライブラリの入手

「WonderWitch 用 WonderGate 対応プログラム開発キット β版」はWonder Witch サポートサイトでダウンロードできます（無償。会員登録が必要です）。ダウンロードには「使用許諾契約書」に同意する必要がありますので，しっかりと読みましょう。

アーカイブには以下のファイルが含まれています。

○ “wg.il”ワンダーゲート対応のインダイレクト・ライブラリ（IL）
○ “wg.h”ヘッダーファイル
○ “wgsock.h”ライブラリの関数をまとめたヘッダーファイル（この記事では使いません）

動作原理

ワンダーゲートはインターネットにPPP（Point to Point Protocol）で接続します。接続できるのはNTT ドコモのデジタル携帯電話です。

アクセスポイントは「mopara」，または指定のプロバイダを選択できます。プロバイダは大半が対応しています。



▲《画面1》ベーマガサイトもこのとおり（一応）見られます

《第1表》インダイレクト・ライブラリ“wg.il”の関数一覧

WgILを初期化する	int far init()
WgILを終了する	void far bye(int bps)
ソケットをオープンする	int far socket(unsigned char far *buf, int tcp)
ソケットをクローズする	int far close(unsigned char far *buf, int socket)
ホスト名からホストエントリ(IPアドレス)を取得する	int far gethostbyname(unsigned char far *buf, char far* name, unsigned char far* iaddr)
リモートホストに接続を行う	int far connect(unsigned char far *buf, int socket, int port, unsigned char far* iaddr)
PPP設定を行う	int far ppp_setup(unsigned char far *buf, char far* user, char far* pass)
PPP接続を切断する	int far ppp_disconnect(unsigned char far *buf)
PPP接続をする	int far ppp_connect(unsigned char far *buf, char far* phone)
電話機状態を得る	int far phone_status(unsigned char far *buf)
ワンダーゲートの電源ON またはバージョンを調べる	int far version(unsigned char far *buf)
ワンダーゲートの電源OFF	int far poweroff(unsigned char far *buf)
一行データ受信(CR+LFを検出)	int far read_line(unsigned char far *buf, int socket)
データ受信	int far read(unsigned char far *buf, int socket)
データ送信	int far write(unsigned char far* buf, int socket, int buflen)

使用方法

インダイレクト・ライブラリの使用方法です。まずソースの先頭でヘッダファイルをインクルードします。

```
#include "wg.h"
```

つぎに関数テーブルとなるグローバル変数を定義します。

```
WgIL wgIL;
```

そして，インダイレクト・ライブラリを参照します。ファイル名が“wg.il”なので，引数は“@wg”と記述します。

```
result = ilibIL->_open("@wg", (IL far *)&wgIL);
```

ライブラリ内の関数を呼び出すときは“wgIL.関数名( )”と記述します。たとえば“init( )”を実行したい場合は

```
bps = wgIL.init( );
```

となります。これら関数の呼び出しはすべて“wgsock.h”で定義済みです。そこで,

```
#include"wb_sock.h"
```

をインクルードして，

```
wg_init( );
```

と記述してもかまいません。“wgIL.～”として使うか，“wb_～”関数を使うかは製作者が好きに選べます。

制御できるプロトコルは「TCP」と「UDP」の2種類です。TCPはコネクト(connect)で接続を確立してから通信します。TCPで接続する場合，以下の手順で関数を呼びます。複雑に見えますが，IL初期化～IL終了処理，PPP接続～PPP切断，ソケットオープン～クローズ，という処理が重なったものです。

①ILの初期化（init）
②PPP設定（ppp_setup）
…アクセスポイントへの接続方法を設定します。
③PPP接続（ppp_connect）
…携帯電話でインターネットへ接続します。PPP接続するためには，「電話番号」「アカウント」「パスワード」の三つが必要です。
④ホスト名からホストエントリを取得(gethostbyname)
…ホストエントリ（IPアドレス）を取得します。
⑤ソケットの作成（socket）
…バッファサイズを与えてソケットを作成します。
⑥ホストに接続（connect）
…相手のホストに接続します。接続するためには相手のIPアドレスとポート番号を与える必要があります。
⑦送信または受信（read／write）
…データの送受信を繰り返します。データは1回で127バイトずつしか送受信できません。
⑧ソケットの切断（close）
⑨PPPの切断（ppp_disconnect）
⑩ILの終了（bye）

サンプルプログラム

β版に付属する“wgget.c”は，サーバから画像ファイルを読み込んでワークエリア（/ram0）に保存するというサンプルプログラムです。自分のプロバイダに合わせて“setup.h”の以下の値を書き換えることができます。

```
#define USER "user@isp"
#define PASS "password"
#define PHONE "000-1111-2222"
```

サンプルでは，処理の最後に読み込んだ画像データを画面表示するために“imager.fx”を実行しますので，あらかじめプログラムエリア（/rom0）に“imager.fx”を入れておくといいでしょう。

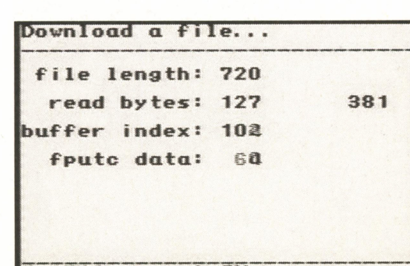
このサンプルはWEBサーバに対して接続を試みます。WEBサーバとの通信方法はWEBブラウザが採用している「HTTP」です。サンプルではワンダーゲートは以下のようなコマンドを送信します。

```
"GET http://wonderwitch.qute.co.jp
/banner10g.fr HTTP/1.0"
```

「GET」が受信コマンド，「http://～」が受信したいデータのURL，「HTTP/1.0」がプロトコルを意味しています。GETコマンドを送ると……

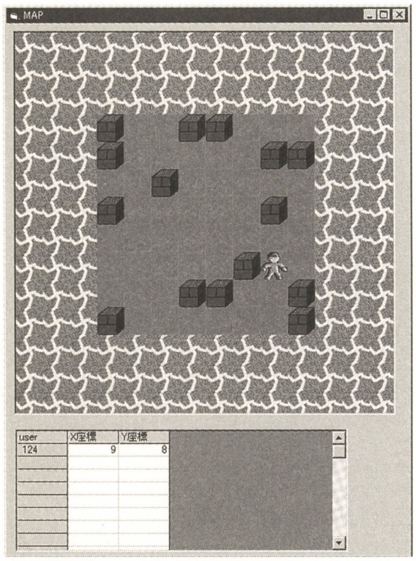
```
HTTP/1.1 200 OK
Date: Thu, 14 Jun 2001 04:27:56
GMT
Server: Apache/1.3.12 (Unix)
ApacheJServ/1.1.2
Last-Modified: Tue, 12 Jun 2001
17:26:10 GMT
ETag: "2f13d-2d0-3b2650b2"
Accept-Ranges: bytes
Content-Length: 720
Connection: close
Content-Type: text/plain
Content-Language: fr
(以下バイナリデータ)
```

……というデータを受け取ります。各行には改行コード（0Dh，0Ah）が付いています。画像データは“Content-Language: ”の後に送られてくるバイナリですから，それ以前のヘッダは飛ばして保存します。ワンダーゲートは一度に127バイトしかデータを受け取れませんので，ファイルへの追記を繰り返し，一つのファイルに結合させていきます（画面2）。

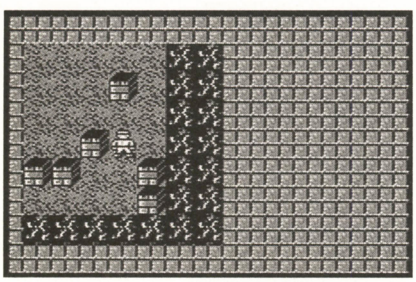


▲《画面2》付属のサンプルの実行結果。WEBサーバから，ある画像をダウンロード中

すべてのポートを自在に扱えますので，応用すればFTPサーバやSMTPサーバとの通信できそうです。



▲《画面3》VisualBasicとWinSock.ocxで作ったサーバ側のプログラム



▲《画面4》サーバから配信されたマップの表示に成功！

ネットワークゲームに挑戦！

ワンダーゲートにはサーバとなって相手からの接続を受ける（accept）機能がありませんので，直接ワンダーゲート同士のTCP/IP通信はできません（UDPなら可能）。しかし，ダイアルアップのためIPアドレスが変動します。これでは相手のIPアドレスがわからず，送信ができません。そこで，パソコンで「対戦サーバ」を作り，それが常にデータを仲介する必要があります。

サーバ側のプログラムはWinSock.ocxを使ってVisualBasicで作りました。2000年9月号の「VB研究所」で紹介した「ネットワークゲーム」のサーバ部分を独立させました（**画面3**）。画面データを確実に更新させるために，送信後は必ずマップデータを受け取るように通信手順を変更しました。あと，送信データの最後につく改行コードに対応させました。



▲《写真2》ワンダーゲート片手に壮絶なデバッグ中の写真

通信に失敗しながらデバッグを繰り返して（電話代を浪費しながら）遂に，完成。ワンダースワンにサーバと同じマップが表示されました（**画面4，写真2**）。サーバには16人まで同時接続ができます（未確認です）。

最後に

ワンダーゲートの入手の難しさ，そして，かさむ電話代と，デバッグのしにくさを考えると，ワンダーゲートは「狭き門」です。しかし，インターネットの用途は無限大ですので，ユーザの方は検討してみてはいかがでしょう。

ワンダーゲートをプレゼント!!!

BM投稿プログラムコーナーにワンダーウィッチ用プログラムを送ってくれた方から抽選で1名様に「モバイルワンダーゲート」をプレゼント。締切は8月末まで！



《リスト1》ネットワークゲームの実験用プログラムリスト“netgame.c”（WonderWitch用）

```
/* ワンダーゲートを使ったネットワークゲームの実験        by 松原拓也 */
/*パソコン側でサーバ用プログラムを実行させる必要があります。*/
#include <stdio.h>
#include <sys/bios.h>
#include <sys/fcntl.h>
#include <stdlib.h>
#include <string.h>
#include <wg.h>

/*キャラクターパレットナンバー*/

#define CPN_TEXT 0 /*テキスト*/
#define CPN_MAP  1 /*スクリーン1(背景)用*/

#define CHR_PIX   8  /*1キャラクター1辺のピクセル数*/
#define ICONW 2  /**/

#define CHN_MAP   0   /*キャラクター番号*/
#define CHN_RENGA 16  /*キャラクター番号*/
#define CHN_MSG   17  /*キャラクター番号*/

#define MAP_X 1
#define MAP_Y 2
#define MAP_W 14 /*幅[キャラクター数]*/
#define MAP_H 14 /*高さ[キャラクター数]*/

#define MSG_X 1
#define MSG_Y 1
#define MSG_W 22 /*幅[キャラクター数]*/
#define MSG_H 1 /*高さ[キャラクター数]*/
```

```
#define IPLEN 4 /*IPアドレス長さ*/
#define BUFSIZE  256 /*バッファサイズ*/
#define PROTCOL_TCP  0   /* プロトコル=TCP        */

/*----------------------------------*/
#define USER ""  /*PPP接続、ユーザー*/
#define PHONE   "" /*PPP接続、電話番号*/
#define PASS "" /*PPP接続、パスワード*/

#define SERVER "***.******.co.jp"   /*サーバのURL*/
#define PORT 5001


static WgIL wgIL;   /*WGインダイレクト・ライブラリ構造体*/
static unsigned char cArry[BUFSIZE];   /*バッファ*/
static unsigned char cIpAdr[IPLEN];   /*IPアドレス*/
static int iBps;
static int iSocket;  /*ソケット番号（ハンドル）*/

/*---------------------キャラクターデータ*/
char cChrDat[]={
   0x77,0x88,0xF9,0x06,0xBE,0x41,0xEF,0x10,0xFB,0x04,0xDC,0x23,0xFF,0x00,0xEB,0x14
  ,0x9D,0x62,0xED,0x12,0xB7,0x48,0xFD,0x02,0x7F,0x80,0xFF,0x00,0xF7,0x08,0x7F,0x80
  ,0xDE,0x21,0xB9,0x46,0xEE,0x11,0x7F,0x80,0xFE,0x01,0xBB,0x44,0xCF,0x30,0xBC,0x43
  ,0xBF,0x40,0xED,0x12,0xFB,0x04,0xBB,0x44,0xBD,0x42,0x56,0xA9,0xDF,0x20,0xFB,0x04
  ,0x70,0x8F,0xE0,0x1F,0xC0,0x3F,0x80,0x7F,0x00,0xFF,0x6D,0xD3,0x7D,0xC3,0x00,0xFF
  ,0x00,0xFF,0x02,0xFD,0x04,0xFB,0x08,0xF7,0x11,0xEE,0x62,0x9D,0xE4,0x1B,0x08,0xF7
  ,0x7F,0xE0,0x7F,0xC0,0x7F,0xE0,0x00,0xFF,0x7D,0xE3,0x7D,0xC3,0x7D,0xF3,0x00,0xFF
  ,0xE2,0x1D,0xE4,0x1B,0xE8,0x17,0x10,0xEF,0xE1,0x9E,0xE3,0x1C,0xE7,0x98,0x0F,0xF0
```

```c
  ,0x00,0xFF,0x01,0xFE,0x00,0xFF,0x02,0xFD,0x30,0xDF,0x30,0xDF,0x1B,0xFC,0x01,0xFE
  ,0x60,0x9F,0xE8,0x17,0x80,0xFF,0x18,0xFF,0x20,0xFF,0x20,0xFF,0x42,0xFD,0x20,0xFF
  ,0x05,0xFB,0x06,0xFB,0x04,0xFF,0x24,0xDF,0x28,0xF7,0x00,0xFF,0x00,0xFF,0x01,0xFE
  ,0x10,0xFF,0x10,0xEF,0x06,0xF9,0x1C,0xFB,0x30,0xDF,0x14,0xFB,0x00,0xFF,0x40,0xBF
  ,0xF8,0x07,0xF7,0x0F,0xF0,0x0F,0xF7,0x0D,0xFB,0x07,0xE0,0x1F,0xDF,0x3E,0xB7,0x6E
  ,0x1F,0xE0,0xEF,0xD0,0x0F,0xF0,0xEF,0xB0,0xDF,0xE0,0x07,0xF8,0xFB,0xFC,0xED,0xFA
  ,0xB7,0x6E,0x80,0x7F,0xB7,0x7F,0xC7,0x3E,0xF6,0x0D,0xE0,0x1F,0xDC,0x33,0xE1,0x1E
  ,0xED,0xDA,0x01,0xFE,0xED,0xDE,0x71,0xEE,0x37,0xF8,0x01,0xFE,0xB9,0x66,0xC1,0x3E
  ,0xFF,0xFE,0xFF,0x01,0xAF,0x51,0xFF,0x01,0xBF,0x41,0xFF,0x01,0x7F,0x81,0xDA,0x25
};

/*--------------------メッセージ表示*/
void subPutMessage(char *pMsg )
{
  /*メッセージウインドウ*/
  text_set_screen(SCREEN1);
  text_window_init(MSG_X ,MSG_Y ,MSG_W ,MSG_H , CHN_MSG);
  text_put_string( 0 ,0 ,pMsg ); /*メッセージ表示*/
}

/*-------------------------------------*/
/* スクリーン表示関数*/
/*引数  int  iScn  */
/*引数  int  sx 開始座標X[キャラクタ数]*/
/*引数  int  sy 開始座標Y[キャラクタ数]*/
/*引数  int  sw 幅[キャラクタ数]*/
/*引数  int  sh 高さ[キャラクタ数]*/
/*引数  int  iChn  キャラ番号*/
/*引数  int  iCpn  パレット*/
int fncScrPut(int iScn ,int sx ,int sy ,int iW ,int iH ,int iChn ,int iCpn)
{
  int x,y;

  for(y=sy ;y<(sy + iH) ;y++) {
    for(x=sx ;x<(sx + iW) ;x++) {
      screen_fill_char(iScn ,x ,y ,1 ,1
             ,iChn | (iCpn  << CFSFT_PALETTE));  /*スクリーン描画*/

      iChn++; /* キャラクター番号+1 */
    }
  }
  return(TRUE);
}

/*---------------------------------*/
/*戻り値：int 文字列の長さ[バイト数]*/
int fncLen(char * pStr)
{
  int iLen=0;

  while(*pStr != 0) {
    iLen++;
    pStr++;
  }
  return(iLen);
}

/*-----------------------------------*/
void  subLoop(int iSocket)
{
  #define TIMEOUT 140

  static char strRx[BUFSIZE];
  static char strTx[BUFSIZE];
  int iKey;
  int iX,iY,iPos;
  int iTout;

  while( 1 ) {
    strcpy(strTx,"MAP");

    for(iTout=0 ;iTout<TIMEOUT ;iTout++) {
      sys_wait(1);
      if((iKey = key_press_check()) != 0)
        break;
    }
    switch(iKey) {
    case KEY_START:
      return;
      break;
    case KEY_DOWN1:
      strcpy(strTx ,"DOWN");
      break;
    case KEY_LEFT1:
      strcpy(strTx ,"LEFT");
      break;
    case KEY_RIGHT1:
      strcpy(strTx ,"RIGHT");
      break;
    case KEY_UP1:
      strcpy(strTx ,"UP");
      break;
    }
    wgIL.write(strTx, iSocket, fncLen(strTx));
    wgIL.read_line( strRx, iSocket );

    iPos = 3; /*最初の「MAP」3文字を無視*/
    for( iY = 0 ;iY < MAP_H ;iY += ICONW) {
      for( iX = 0 ;iX < MAP_W ; iX += ICONW) {

        fncScrPut(SCREEN1 ,iX+MAP_X ,iY+MAP_Y ,ICONW ,ICONW
          ,(strRx[iPos]-'0')*(ICONW*ICONW) + CHN_MAP ,CPN_MAP);

        iPos++;
      }
    }
  }
}


/*----------------------メイン-------*/
void main(int argc, char *argv[])
{
  int iRet;

  lcd_set_color(0x6530, 0xfca8); /* LCD階調データ */
  display_control(DCM_SCR1);  /*画面表示*/
  font_set_colordata(CHN_MAP ,sizeof(cChrDat)/16 ,cChrDat);  /*キャラクター設定*/
  screen_fill_char(SCREEN1 ,0,0,SCREEN_CHAR_WIDTH ,SCREEN_CHAR_HEIGHT
          , CHN_RENGA | (CPN_TEXT  << CFSFT_PALETTE));
  palette_set_color(CPN_TEXT ,0x7140);  /*パレット設定(00,01,10,11 LCDカラー番号)*/
  palette_set_color(CPN_MAP  ,0x0741);  /*パレット設定(00,01,10,11 LCDカラー番号)*/

  iRet = ilibIL->_open( "@wg", (IL far *)&wgIL);
  if( iRet != E_FS_SUCCESS )
    return;

  iBps = wgIL.init();  /*IL初期化*/

  wgIL.version( cArry );
  wgIL.ppp_setup( cArry, USER, PASS );  /*PPP設定*/

  /*PPP接続*/
  if( wgIL.ppp_connect( cArry, PHONE )!= 0) {
    subPutMessage("PPP Connect NG"); /*PPP失敗*/
  }else{
    subPutMessage("PPP Connect OK");

    wgIL.gethostbyname( cArry, SERVER, cIpAdr );
/*    cIpAdr[0]= 0;
    cIpAdr[1]= 0;
    cIpAdr[2]= 0;
    cIpAdr[3]= 0;   IP直打ち用*/

    iSocket = wgIL.socket( cArry, PROTCOL_TCP );

    /*ソケット接続*/
    if( wgIL.connect( cArry ,iSocket ,PORT ,cIpAdr ) != 1) {
      subPutMessage("Socket Connect NG"); /*コネクト失敗*/
    }else{
      subPutMessage("Socket Connect OK");

      subLoop(iSocket);

      wgIL.close( cArry, iSocket ); /*ソケット切断*/
    }
    wgIL.ppp_disconnect( (unsigned char far *)cArry ); /*PPP切断*/
  }
  wgIL.bye(iBps); /*IL終了*/
}
```

※この記事はpreBeta版をもとに作成しています。

携帯電話で動くゲームを作ろう!!

君にもできる「iアプリ」制作



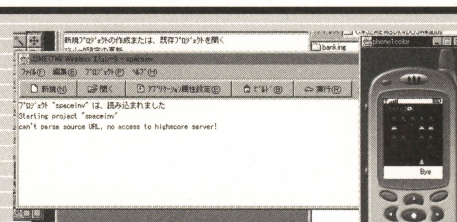
第5弾「iアプリで音楽再生」

今月はiアプリで音楽を再生する方法を紹介します。（松原拓也）

iアプリの開発　先月のおさらいですが，iアプリの開発方法を紹介します。

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

② NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からiアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa）」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA”フォルダに主要なファイルが展開されます。



③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと，その下に“bin”，“res”，“src”フォルダが作成されます。

⑤“res”フォルダに画像（gif）や音楽ファイル（mld），“src”フォルダにはソースファイル（java）を格納します。

⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します（“AppSize”と“LastModified”は自動的に更新されます）。

AppName = (アプリケーション名)
AppVer = 1.0
AppClass = (最初に実行するクラス名)
SPsize = (スクラッチパッドのサイズ)
KvmVer = CLDC-1.0


ｱﾌﾟﾘｹｰｼｮﾝ属性の設定 "cinpact"

ｷｰ	属性値
AppName	チープインパクト
AppSize	3559
AppVer	1.0
PackageURL	cinpact.jar
KvmVer	CLDC-1.0
SPsize	0
AppClass	GameApp
AppParam	
LastModified	Wed, 09 May 2001 19:15:12
UseNetwork	
TargetDevice	
LaunchAt	

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは"res"フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1color”フォルダの“phone1color.properties”を以下のように書き換えます（値は120×130ピクセル時の設定です）。書き換えたら「編集－設定」をメニュー選択して，「デバイス指定」を“phone1color”にします。

```
:
screen.x=27
screen.y=65
:
screen.width=120
screen.height=156
:
icon.up: 44, 138, off
:
icon.down: 44, 146, off
:
screenPaintableRegion.width=120
screenPaintableRegion.height=130
:
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
:
```



⑨「実行」ボタンを押すと，携帯電話エミュレータ上でiアプリが実行されます。

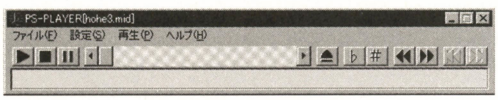
iアプリで音楽再生！

iアプリは「iメロディ」という形式に対応しています。「iメロディ」は1～4トラック4和音（503iは4トラック16和音）の音楽データで，パソコン上にファイルとして保存することができます。拡張子は“mld”です。

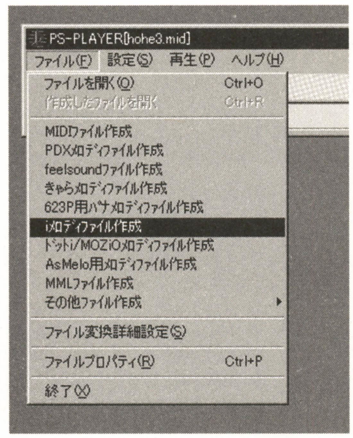
iアプリはMIDIファイルには対応していません。しかし，MIDIをiメロディに変換すればMIDIの曲を聞くことができます。まずMIDIの曲を各種シーケンスソフトや楽器を使って作成します。保存時の拡張子は“mid”です。

MIDIからiメロディへの変換はフリーウエアの「PS-PLAYER v07.78」を使いました。「PS-PLAYER」は作者であるAPOさんのWEBサイト（http://www.asahi-net.or.jp/~wj3a-fji/）の「PHSをちょい研究！」というページで公開されています。

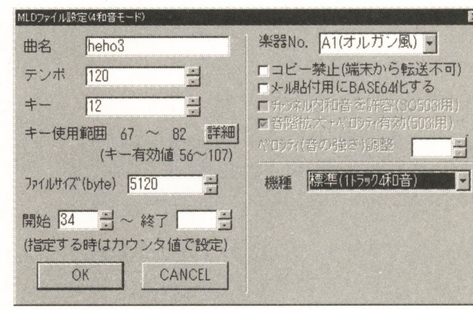
変換方法は「PS-PLAYER」（ps-player.exe）を立ち上げてMIDIファイルを読み込み，「ファイルーiメロディファイル作成」をメニュー選択するだけです。これでMIDI（mid）がiメロディ（mld）ファイルになって保存されます。



▲「PS-PLAYER v07.78」（対応OSはWindows95/98）



▲MIDIをiメロディに変換



▲変換時の設定画面

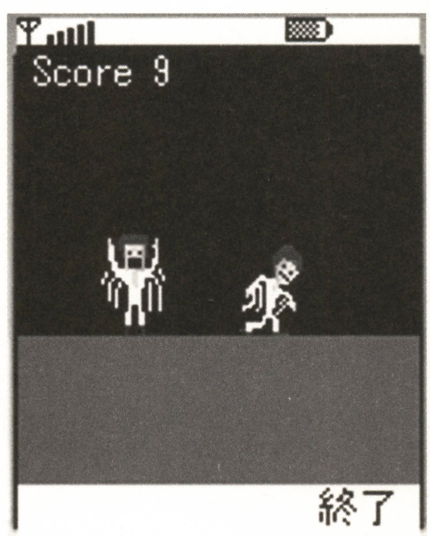
音の出るiアプリのゲーム

『スーパー★スター』

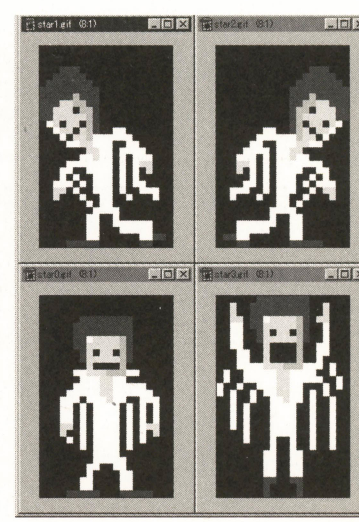
これは，先生の踊りを真似して踊る，昔のベーマガにあった投稿作品をヒントにして作ったゲームです。ゲームの最初にBGMが鳴ります。

操作方法

左･上･右ボタンで体を動かします。何も押さないと直立不動はそのままです。先生の体の動きに合っていないと，ゲーム･オーバーとなります。あと，得点が上がっていくと先生の踊りが速くなっていきます。



◀先生（右側）の動きに合わせて，プレイヤー（左側）が踊るというゲーム。BGMが鳴ります



▲プレイヤーの状態を4枚のgif画像で保存（20×30ピクセル）

《リスト1》"GameApp.java"

```
/*-------------------------------------------------------------*/
/* "GameApp.java"  by 松原拓也*/

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class GameApp extends IApplication
{
    public void start()
    {
        GameScreen screen = new GameScreen(); /*キャンバスの作成*/
        Display.setCurrent(screen);  /*キャンバスのカレント表示*/
    }
}
```

《リスト2》"sound.jam"

```
PackageURL = sound.jar
AppSize = 3989
AppName = sound
AppVer = 1.0
AppParam =
AppClass = GameApp
SPsize = 0
LastModified = Sat, 16 Jun 2001 20:29:30
KvmVer = CLDC-1.0
```

入力方法

ソースファイル（“GameApp.java” “GameScreen.java”）を入力します。適当なmldファイルを用意して“music.mld”という名前で保存してください。あと，以下の組み合わせの画像を描いてgifファイルで保存してください。

“star0.gif”「何も押さないとき」
“star1.gif”「左を押したとき」
“star2.gif”「右を押したとき」
“star3.gif”「上を押したとき」

プログラムの解説

❶複数のイメージ（画像）を，1個の配列として定義してみました。書式がC言語と少々ちがうのですが，これで「ImgStar[]」の配列が定義されます。

❷音楽再生用のオブジェクト「Audio Presenter」を定義します。

❸イメージを配列にしましたので，4枚のgif画像を，一つのforループで読み込んでいます。なお，gifのファイル名はループ回数から作りだしています。

❹「MediaManager」オブジェクトの「getSound」メソッドでmldファイルを読み込みます。音楽データは「MediaSound」に読み込まれます。

❺「setSound」メソッドで音楽データを「MediaSound」から「Audio Presenter」オブジェクトに転送します（この時点で「MediaSound」オブジェクトは破棄できます）。これで「play」メソッドを呼べば，音楽が再生されます。音楽の停止は「stop」です。

❻「drawImage」でイメージを描画します。今回は「プレイヤー」と「先生」という2人を描くのに，一つのforループで処理しています。

❼「先生」の踊りのパターンを乱数で求めます。「PATMAX」が4なので，

《リスト3》"GameScreen.java"

```
/*----------------------------------------------------------------*/
/* 「スーパー★スター」キャンバス "GameScreen.java"  by 松原拓也 */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

public class GameScreen extends Canvas implements Runnable {

    final int TIPW = 30;        /* 常数:キャラの高さ[ピクセル数] */
    final int SEQPLAYING = 0;       /*シーケンス：プレイ中*/
    final int SEQGAMEOVER = 2;      /*シーケンス：ゲームオーバー*/
    final int SEQGAMENEXT = 160;    /*シーケンス：次のゲーム*/
    final int GAME_SPEED = 70;      /*ゲームのスピード*/
    final int PATMAX = 4;  /*踊りパターン数*/
    final int MY      = 0;   /*プレイヤー*/
    final int SENSEI  = 1;   /*踊りの先生*/
    final int PLAYERS = 2;   /*登場人物数*/

    Thread myThread;  /* スレッド */
    Image imgStar[] = new Image[PATMAX];    /* イメージ */  ◀……❶
    AudioPresenter audio;       /*オーディオ*/ ◀……❷
    int[] iX = new int[PLAYERS];
    int[] iY = new int[PLAYERS];
    int[] iP = new int[PLAYERS];
    int iFlat = (getHeight() / 3)*2;  /*地平線のY座標*/
    int iSeq;      /*シーケンス*/
    int iScore,iMiss;   /*スコア、失敗度数*/
    int iNando;  /* 難易度 */
    Random rnd = new Random();   /* 新しい乱数列を作成*/

    public void subInit() {    /*ゲームデータのクリア*/
        int i;
        for(i=0 ; i<PLAYERS ;i++ ) {
            iX[i] = (getWidth() / 3)*(i+1) - (TIPW / 2);   /* 主人公座標 */
            iY[i] = iFlat - TIPW;
            iP[i] = 0;
        }
        iSeq = SEQPLAYING;  /*シーケンス初期化*/
        iScore = 0;    /*スコア0点*/
        iMiss = 100;   /*ミスしていい回数*/
        iNando = 90;    /*難易度*/
    }

    public GameScreen() {    /*コンストラクタ*/
        int i;
        MediaSound ms;
        MediaImage mi;

        setSoftLabel(Frame.SOFT_KEY_2, "終了");

        for(i=0 ;i < PATMAX ;i++) {
            mi = MediaManager.getImage("resource:///star"+ i +".gif"); ◀……❸
            try {
                    mi.use(); /* イメージの読み込み */
            } catch (ConnectionException e) {
            }
            imgStar[i] = mi.getImage(); /* GIF画像読込 */
        }

        ms = MediaManager.getSound("resource:///music.mld"); /*iメロディ読込*/ ◀…❹
        try{
            ms.use();
        }catch (ConnectionException e){
        }
        audio = AudioPresenter.getAudioPresenter();
        audio.setSound(ms);
        audio.play();   /*サウンド再生開始*/ ◀……❺

        subInit();  /*ゲームデータのクリア*/
```

```
        myThread = new Thread(this); /* スレッド作成 */
        myThread.start();  /* スレッド開始 */
    }

    public void paint(Graphics g) {  /*ペイントメソッド*/
        int i;
        g.lock();  /* 画面のロック */

        g.setColor(g.getColorOfRGB(0, 0, 0));
        g.fillRect(0, 0, getWidth(), iFlat ); /*黒い空*/

        g.setColor(g.getColorOfRGB(120, 100, 64));
        g.fillRect(0, iFlat , getWidth(), getHeight()-iFlat ); /**/

        for(i=0 ;i < PLAYERS ;i++) {
            g.drawImage(imgStar[iP[i]] ,iX[i] ,iY[i]);   /* 描画 */ ◀……………… ❻
        }

        g.setColor(g.getColorOfRGB(255, 255, 255));
        g.drawString("Score "+iScore, 5, 12);

        if(iSeq > SEQGAMEOVER) {
            g.drawString("先生「だめだよー」", 10, 40);
        }
        g.unlock(false);  /* ロック解除 */
    }

    public void run() {  /* スレッド実行 */
        int iKey;

        for(;;) {
            try {
                Thread.sleep(GAME_SPEED);
            } catch(InterruptedException ignore) { }


            iKey=getKeypadState();        /* 主人公 */
            iP[MY]=0;  /*踊りパターンなにもせず*/

            if(iKey==(1 << Display.KEY_LEFT))
                iP[MY]=1;     /*踊りパターン*/

            if(iKey==(1 << Display.KEY_RIGHT))
                iP[MY]=2;     /*踊りパターン*/

            if(iKey==(1 << Display.KEY_UP))
                iP[MY]=3;     /*踊りパターン*/

            if( Math.abs(rnd.nextInt() % 100) > iNando) {
                iP[SENSEI] = Math.abs(rnd.nextInt() % PATMAX); ◀……………… ❼
            }

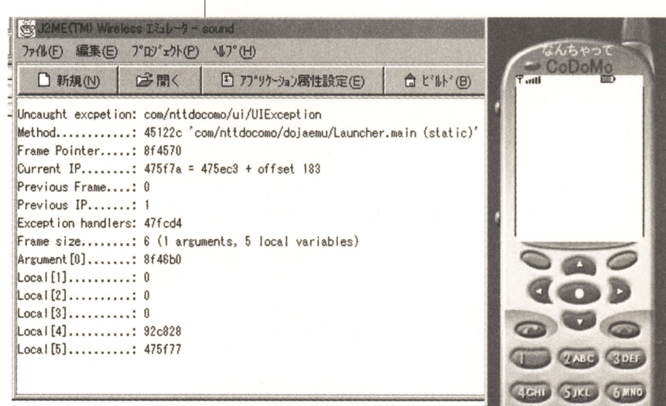
            if(iSeq == SEQPLAYING) {  /*ゲーム中*/
                if(iP[MY]==iP[SENSEI]) {  ◀……………… ❽
                    iScore++;      /*スコアをプラス*/
                    if((iScore % 10)==0)
                        iNando--;  /*難易度プラス*/
                }else{
                    iMiss --;
                    if(iMiss <= 0)
                        iSeq = SEQGAMEOVER;
                }
            }else{
                iSeq++;
                if(iSeq >= SEQGAMENEXT)  /*ゲームオーバー*/
                    subInit();  /*ゲームデータのクリア*/
            }
            if(iKey==(1 << Display.KEY_SOFT2)) {
                IApplication.getCurrentApp().terminate(); /*終了*/
            }
            repaint(); /*再描画*/
        }
    }
}
```

「Math.abs(rnd.nextInt() % PATMAX)」だと，0〜3の乱数を返します。

❽「プレイヤー」と「先生」の踊りが合っているかのチェックです。合っていれば得点を加算します。得点が上がるたびに難易度を上げていきます。

注意

「DoJa」付属のエミュレータはmldファイルを再生することができずにエラーになってしまいます。このためデバッグ中は代わりにMIDIファイル（mid）を指定して音を鳴らし，実機で動作させる前にmldに切り替えるという方法があります。なお，「i-JADE」ではきちんとmldファイルが再生されます。



▲「DoJa」ではmldが再生できずにエラー（Exception）になってしまう

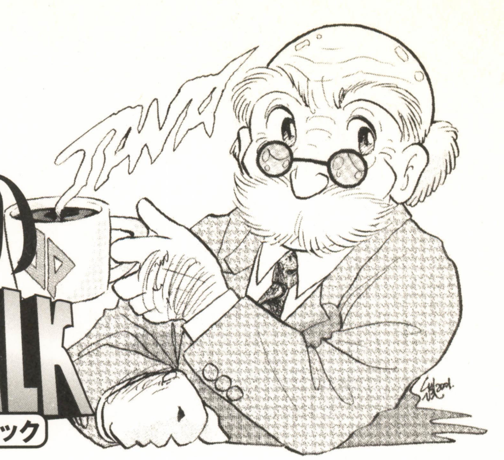


▲先生と踊りが合わないとゲーム・オーバー！

超 初心者向けJava入門

魁!! Dr.Jの JAVA-WALK

ジャバウォック



第3回 プログラムを実行してみよう！

とにかく実行してみよう！

やぁ，諸君！ワシがJAVA塾・塾長邪罵島平八郎…じゃなかった，Dr.Dの従兄弟こと，Dr.J(ドクトルジェイ)である！最近ワシのことを架空の人物だとかなんだとか疑う輩がおるようじゃが，ワシを信じてプログラムを作るのじゃ！

…さて。夏休み，いかがお過ごしじゃろうか？宿題は早めにすませて，投稿プログラム作りに暑い，いや熱い青春の1ページを捧げるのじゃあっ！う〜ん，若いってホントいいのぉ〜。ワシもまだまだ若いぞ！なんたってハタチじゃからな！おっと，いや，その，カンレキから数えてってカンジかの？ホッホッホッ！

前回はJAVAプログラミング環境のインストールに成功した！いよいよプログラムの実行じゃ！何はともあれ，まずは動かしてみる！理屈はわからんでもとにかく動かすことが重要じゃ！自分で書いたプログラムが動けば楽しい！これぞプログラミングの極意であり，やってみないことにはワカランのである！

まずはソース・ファイルを作るのだ

まずはソース・ファイルの作成じゃ！ソースとはすなわち源(みなもと)，その名のとおりプログラムのもとになるものだ！もとになるプログラム「ソース・プログラム」の書かれたファイルのことを「ソース・ファイル」と呼ぶのである！とにかくソース・ファイルさえ作れれば，プログラミングなんぞ8割方できたと思って間違いない！

ナニ，何を使って作るかって？うむ，良い質問である！ソース・ファイルはテキスト・ファイルでなければならない…テキスト・ファイルとは何か？混じりけなしの文字だけで作られたファイルじゃ！市販のテキスト・エディタでもワープロでも，OSについてるメモ帳でも，どんなツールを使っても構わんのだが，保存するときの形式を「すべてのファイル(＊.＊)」とか「テキスト文書」とか「テキストのみ(＊.txt)」にすることを忘れないでくれたまえ（まぁワープロ文書の形式にはしないように)。

では，ここでは誰でも持っておる"メモ帳"を使って作成しよう！

[スタート]-[プログラム]-[アクセサリ]-[メモ帳]

そのままこのソース・プログラムを打ち込んでくれたまえ！

```
public class Helloworld {
   public static void main(String args[]) {
       System.out.println("Hello World");
   }
}
```

大文字・小文字の間違いはないかな？OKだったら，テキスト・ファイルとして保存しよう！

→[ファイル]-[名前をつけて保存]

おっと，ここで保存する場所を，どこか新しく決めておこう。

→開かれている「名前をつけて保存」画面の，[保存する場所]で（C:）を選ぶ

→右側のアイコン[新しいフォルダの作成]を押して新しいフォルダを作成する

→反転表示の「新しいフォルダ」に，新フォルダ名「javawalk」を打ち込む

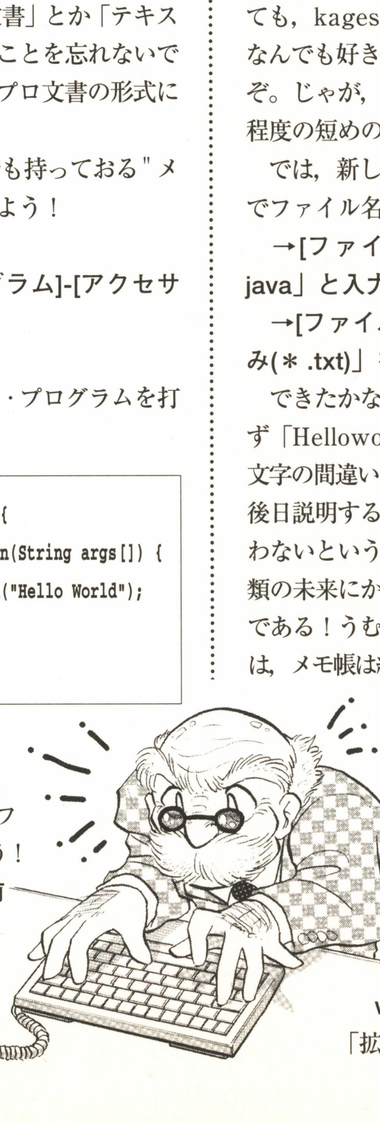
これで，Cドライブにjavawalkという作業用のフォルダができあがったわけじゃ。もちろんここはjavawalkでなくても，kagesanでもtsugumichanでもなんでも好きなフォルダ名にして構わんぞ。じゃが，後々のことを考えて8文字程度の短めの名前が望ましいの。

では，新しいフォルダができたところでファイル名を入力して保存しよう。

→[ファイル名] に 「Helloworld.java」と入力し，

→[ファイルの種類] を「テキストのみ(＊.txt)」を選択して，[保存]

できたかな？ここではファイル名は必ず「Helloworld.java」に，大文字・小文字の間違いなく打ち込んでくれたまえ。後日説明するが，このファイル名を間違わないということについては，じつは人類の未来にかかわる重大な意味があるのである！うむ，ちょっと大げさかな？では，メモ帳は終了させて構わんぞ。で，格納したフォルダjavawalkを確認しよう。お節介なことに，なぜかHello world.java. txtというファイル名になってしまっておるので，これはファイル名を修正しておこう。



→クリックしてHello world.javaに修正

「拡張子を修正すると…」とか，



不安になるようなメッセージが出てくるが，構うこたぁないので [はい] にしてしまおう。気の利いたエディタならこんな面倒はないのじゃが…とほほ。

実行あるのみなのだ！

では，早速実行してみよう！…って，第1回で説明したとおり，ソース・ファイルはコンパイルして，実行ファイル（JAVA プログラム）を生成しなければならない！方法は簡単じゃ。

→[スタート]-[プログラム]-[MS-DOS プロンプト] で MS-DOS 画面を表示する

む，Windows のディレクトリにおるな。さっきの作業フォルダにするために，cd c:¥javawalk と入力しよう。

```
C:\WINDOWS>cd c:\javawalk
C:\javawalk>
```

よおーし。では，実行である！ Javac Helloworld.java と入力じゃ！

```
C:\javawalk>Javac Helloworld.java
コマンドまたはファイル名が違います.
```

あり？なんだ，これは！？間違ったのであるか？そんな，世界征服の野望が…なんてことはないのじゃ！ここで必要なのが，先月の宿題にあった秘密の呪文なのである！何かって？「パス」じゃ！パスの設定をするのじゃ！「パス」とはナニかって？ここでいうパスとは，"Path"であり，道筋とかそういう意味なのだ！早い話が，javaのコンパイラのプログラムがどこにあるかわからんかった，ということなのだ。

したがって，どこにコンパイラがあるかを教えてあげればよいのである！ set PATH というコマンドを使って，set PATH=c:¥jdk1.3.0_02¥binと打ち込めば良いのじゃぞ(jdk1.3.0_02は，前回インストールしたフォルダ名～MS-DOSプロンプトなので「ディレクトリ名」と呼ぶこともあるぞ)。

```
C:\javawalk>set PATH=c:\jdk1.3.0_02\bin
```

では，もういちどコンパイラを実行じゃ。

```
C:\javawalk>Javac Helloworld.java
C:\javawalk>
```

おお！ナニも文句を言われない！と，いうことはうまくできたってことじゃろうか？お，確かにフォルダには新しいファイル Helloworld.class ができておる。実行は，確かそのまんまJava Helloworldと打ち込めばよいのじゃったな。

```
C:\javawalk>Java Helloworld
Hello World
```

おおおお！なんか出たぞ！「Hello World」って表示された！これはソース・プログラムの中の「System.out.println ("Hello World");」という部分にあった文字ではないか！ファイル名なんかは「Helloworld」で，つづりがちがうよな…ってことは，諸君，喜びたまえ！苦節丸2か月，3か月目にして，遂にJAVAプログラムを動かすことができたのじゃ！うぉぉ！感動じゃぁぁ！

ほんとに動いたのか確かめてみよう

ナニ，ほんとかどーか怪しいじゃと！？よろしい，ではさっきのソース・ファイルを変更してみよう。このプログラムは，じつは" "（ダブルクォーテーション）で囲んだ文字（「文字列」と呼ぶのだ）を表示するだけのごくシンプルなものなのである。したがって，この文字列を書き換えてやれば，表示される文字もまた，変わるはずであるのだな。では，つぎのように変更しよう。

```
public class Helloworld {
 public static void main(String args[]) {
  System.out.println("mycom BASIC maga-
zine");
     }
  }
```

ではコンパイル，実行してみよう。

```
C:\javawalk>Javac Helloworld.java
C:\javawalk>Java Helloworld
mycom BASIC magazine
```

む！確かに変更された！間違いない，ラピュタは本当にあったんだ…じゃなくて，これでよいのじゃ！やった！カンペキじゃ！どうじゃな，「おまえはすでにプログラミングしている」って感じじゃな！

さて，今回のjavaプログラムでは，変更して構わない，表示するための文字列の他に，いくつかの単語が使われておる。public, class, static, void, main,String, args, System.out.println というところじゃな。これらはたとえば，

public：クラス修飾子／class：クラス宣言／static：（この場合）メソッド修飾子／void：型宣言／main：プログラムに必ず必要なメイン・メソッド／System.out.println：文字列表示…

とかいったところなのじゃが，これらはすべて「予約語」と呼ばれておる。このへんはすべて次回以降でゆっくり説明していくから，お楽しみに，じゃぞ。もちろん，予習できる勇敢なる読者諸君はがんばってくれたまえ！

では来月までの課題じゃ

予習ついでに，今回ソース・ファイル名をソース。プログラム中の"class"に続く"Helloworld"と合わせておる。これを違えるとどうなるかな？ゼヒ，試してみてほしい。そして，なぜそうなるか，を考えてみてくれたまえ！

夏休みはまだまだ長い！暑さしのぎにプログラミングもオツなもんであろう！若さ爆裂，暑さ猛烈じゃ！ではまた来月，To Be Continued じゃ!!

Catlikeの もっと身近にプログラミング！

第4話 タイピング・ゲーム編 その3 ローマ字入力実践プログラム！

日本語の入力方法の統計を調べてみたら，若い人ほどローマ字入力派が多いそうです。Webページのurlなど英文字を打つ機会が増えたこの頃は，ローマ字入力のほうが便利なのかもしれませんね。

あなたは何派?

かき氷が美味しいのです。シロップはやっぱり甘ぁい「いちごみるく」で決まりだと思うのですが，しゃきーんと心地よいレモンも捨てがたいと思うのです。冷凍庫で氷を大量生産しつつ，もう15年以上も使っているであろう，ぺんぎんさんのかき氷機でしゃかしゃかと氷を削るのが幸せな日々。ああ，やっぱり夏はいいですねえ。この「かき氷タイム」だけは家中のクーラーを止め，窓は力いっぱい大全開！　体中で夏を受け止めながら，流れる汗といちごみるくを楽しんでいたいのです。

適度な刺激に体をリフレッシュしたら，さあ，今度は頭を動かす番。プログラミングはとても知的なゲームとして，夏のひとときを過ごすには最適なのだと思うのです。

ローマ字処理プログラム大掲載!

さてさて今にも海に山にと飛び出していきたい気分ですが，ぐっと我慢してプログラミングのお話をさせていただきましょう。

お店の陳列棚を見回すと，以前にも増して，タイピング・ゲームが数多くリリースされてきていますね。その内訳といえば，キャラクターにあやかったものも多いですが，オリジナルの内容で勝負してきているものも見受けられます。もうこれは，タイピング・ゲームというジャンルが一般的に受け入れられてきた証拠！　とはいえ，比較的新しいジャンルということもあり，実際何年もお仕事でプログラムを組んできた人たちからすると，「きっと作れば組めるんだろうな，でも仕事でまだ組んだことないし，僕らみたいな古いプログラマーが介入することもないんだろうな」などと考えている人もいるそうで，いやはや，これはもしかしたら新人プログラマーさんにとってもチャンスと言えるのかも，なんて思ったりしています。

そんなわけかどうかはともかく，タイピング・ゲーム編最終回を飾るのは，すぐに実践に使えてしまえそうなサンプル・プログラムです。テーブル法を採用しているので，入力方法を変更したり，対応文字を追加したりするのも簡単解決。今回のサンプルはローマ字入力オンリーですが，これをカナ入力に変更したりするのも簡単にできるようになっています。

いやその，実際に作ってみたら意外にプログラム長くなっちゃって，編集さんに聞いたらリストだけで2ページ取っちゃうよって言われて，カナ部分を作るに作れなくなった現状があるのは秘密の事情なのです。カナ部分は，キーコードが絡んでくるので，マシンやOSに特化したプログラムになってしまうのもありますしね。

筆者紹介

catlikeホームページ
http://catlike.marchen.to/
「いつか,何処かで。」専用ページ,多数のアクセスありがとうございます～。そろそろ新しい動きが見えてきますよ。チェックよろしく!

文:まかべひろし
夏と冬はこそこそっと海のほうに出かけることにしてるのです。今年も行きます。まぁそれはともかく,前回言ってたPlayStationのお仕事が終わってホッとしています。そろそろcatlikeとしての作品も発表していきますね。お楽しみに!

絵:桜部のとす
前回に引き続きまた電車描いてみました～。何線だかわかります? ちなみに今はもう走ってない車輌ですけど(^^;　あとはPSO Ver.2が楽しいです。作業進まなくてピンチ。

カギはテーブル抽出!

では早速，**リスト1**を見ていきましょ。ざっと眺めると，おお，リストの大半がテーブルでできていることに気づくでし



ょう。この，「ひらがな数文字」と「英文字」の組み合わせでできているテーブルが，文字のローマ字変換表。ローマ字入力のルールを決める部分。pszRingKeyTblxxxテーブル部分で，対応する「ひらがな文字」および，どう入力したらその文字と判定するのか，という「入力英文字」を指定しています。

たとえば，

```
const unsigned char
pszRingKeyTbl0222 [ ] [8] =
{"し","SI","SHI",""} ;
```

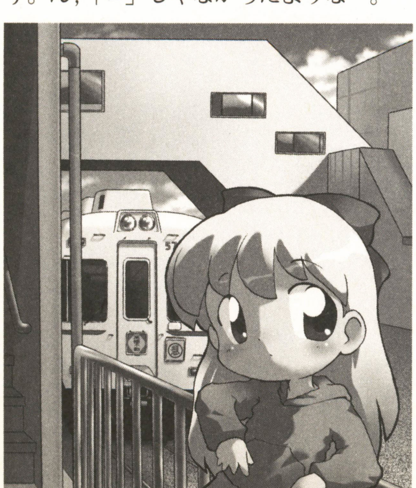
という1行では，「し」というひらがなを入力させるのに，「SI」と「SHI」という2種類の入力方法を指定しています。最後の""は，これで終わりというマークなのです。

また，同じ「し」から始まる言葉として，

```
const unsigned char
pszRingKeyTbl026[][8] = {
"しゃ","SYA","SHA","" };
```

というふうに「しゃ」というひらがなを登録しています。ローマ字の表記法としてありえるものを，すべてこのように定義することで，入力のルールを決めているのです。このおっきな配列群を「ローマ字テーブル」と呼んでおきましょう。

さて，タイピング・ゲームではあらかじめ，入力すべき文章が決められているのが普通です。「ザコとはちがうのだよ，ザコとは」とか，画面に表示された文章を正しくタイピングできるかどうか，というゲームなので，これは当然と言えます。ん，「コ」じゃなかったような…。



まぁとにかく。そうなると，ゲームを作る側で決めることができる部分は，この「問題文」部分になりますよね。ユーザーの入力は不特定要素なので，それをもとに解析処理を行うのは適当とは言えません。決定要素である問題文をもとに解析処理を行ったほうが，処理量からいっても有利ですし，なによりプログラムが簡単です。

たとえば，問題文が「シャム猫」だとしましょう。ひらがなで定義すると「しゃむねこ」。これが解析処理を行うターゲットとなります。この問題文をローマ字入力で打ち込むのに，あなたならどうキータイプすることを考えますか？

まず1文字目である「し」に注目しますよね。「し」をローマ字で構成する入力要素は「SI」と「SHI」です。このいずれかを入力することで，1文字目が確定し，つぎの「ゃ」の入力に移ります。これはローマ字入力なら「XYA」とか「LYA」とか打ちますね。これを問題文が終了までループすれば，問題文の打ち込みが完了しますね。

しかし，この問題文の場合，「しゃ」から打ち込み始めることも可能です。というか，どちらかというと，ローマ字表記としてはこちらが正しいのだと思います。「しゃ」を構成する入力要素は「SYA」および「SHA」。これもローマ字テーブルにあるとおりで，この入力を完了すると，一気に「しゃ」という2文字が確定し，つぎの「む」の入力に移ります。「ゃ」を単独で打つ必要はありません。

タイピング・ゲームとしては，このどちらの例も容認しなくてはなりません。これが頭の悩ませどころなのですが，解析処理プログラムでは，これをテーブル抽出という方法で解決します。

プログラムを組むには，データの注目点を決めなくてはいけません。プログラムの基本は「人間の思考のトレース」ですから，まず，ぼくらが打ち込むときと同じように，問題文の頭に注目させます。「しゃむねこ」の頭から，ローマ字テーブルと一致するものを探し出させるわけです。

ローマ字テーブルのほうは，一つのひらがなに対し，複数の入力要素が入っています。これでは実際の検索のときに不便ですので，一つのひらがなに対し一つの入力要素という形で分解し，別のテーブルに移してやります。これは，後の処理を簡単にするとともに，別テーブルに移すときに，本来のローマ字テーブルを改造して登録することができるという二つの利点があります。とくに後者は，「っ」の入力方法である子音の重ねを行うときには欠かせません。

これを繰り返すことで，ローマ字から必要なテーブルのみを抽出した「入力テーブル」を作成しています。この入力テーブルを作成するタイミングは，一番最初の入力開始時と，1ワード打ち込み終わったときになります。

入力テーブルをつぶしていくんだ

入力テーブルが出来上がったら，入力を待ち，キーボードから何か文字が入力されたときに，作った入力テーブルと同じ文字が入ったかどうかを判定していきます。「しゃむねこ」の例ならば，最初の入力は「S」だけですね。**リスト1**では，cKey変数に入力されたキーデータを引っ張ってきています。これを，入力テーブルと比較し，入力の成否判定をしてやります。キーデータと一致したテーブルは，今後の入力にも使うので残しておきますが，それ以外のテーブルは今後の入力には使いませんので，削除予約マークをつけておきます。

マークだけにしておく理由は，もしかしたら入力が成立しない，ミスタイプかもしれないからです。マークだけなら解除も簡単ですからね。

で，キーデータが一つでも入力テーブルと一致した場合，削除予約マークがついたテーブルは削除します。削除を行わないと，つぎの入力時に関係のないテーブルで反応してしまいますので，忘れないようにしなくてはいけません。反対に，キーデータが入力テーブルと一致しなかった場合は，予約マークはすべて消して「入力失敗」として処理を行います。ゲームによってはミスのエフェクトなどを行う場所ですね。

このように入力テーブルをどんどん削除していくと，ユーザーが入力しようと思っている言葉が絞られていきます。こ

の「削除」という処理を行う関係もあり，元のローマ字テーブルをそのまま入力テーブルとして使うよりは，ローマ字テーブルから入力テーブルを生成するほうが処理が簡単なわけです。

こうして1ワード分入力が終わったら，そのワードの長さ分だけ，問題文を読み進みます。「しゃ」と打ったのならば，2文字分読み進んで，つぎに入力するのは「む」から，というわけです。問題文がそれで終わりならば入力完成となるわけですが，まだ問題文が残っている場合は，新しい場所からの新しい入力テーブルを作成しなくてはなりません。これを繰り返していけば，いつかは問題文が完成！入力処理プログラムも完成です。

でも特例処理があるんだな

しかしですね，ローマ字入力には二つの特例処理があります。

まず一つは「っ」の入力。たとえば「だった」と入力するときは，「っ」の後の言葉の子音を重ねて「DATTA」と入力するのが普通でしょう。これはローマ字テーブルをさらに追加することで対応は可能ですが，それにしてはちょっと分量が多すぎます。「っ」が手前につく可能性はかなりあるわけですから，単純に考えただけでおおよそ倍のテーブル量になってしまいます。

そこで「っ」が来たときのみ，そのつぎの文字に読み進み，そこから子音を重ねた形にプログラム側で編集を行って，入力テーブルを生成してみました。

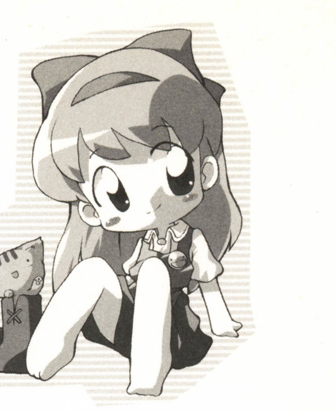
元のテーブルから新たにテーブルを作成するというのは，転送の間でプログラムによる編集が行えるのも利点の一つでしょう。これならばわざわざローマ字テーブルで用意しなくても，子音を重ねた入力の処理を行うことができます。

もう一つの特例処理は「ん」の入力です。「ん」は基本的に「N」1回の入力ですが，「NN」のように2回の入力でも確定しなくてはいけません。この場合，単純な解決方法としては，通常「N」1回で確定してしまい，もしつぎに「N」が押された場合は，特例でミスとしてカウントしないという方法が考えられます。本来の日本語入力処理プログラムとしては「N」のつぎの文字が押されるまで「ん」を確定しないというのが正しい方法のような気もしますが，最後に「ん」が来た場合の処理に困ってしまうということもあり，2回目の「N」はミスとしてカウントしないという方法で今回のサンプルは実装してみました。

ただこの方法にはバグが残されていて，「ん」の後に「なにぬねの」系の「N」で始まる文字がくる場合はちょっと入力が怪しくなってしまいます。これが問題だと思う人は，自分でプログラムを変更して直してみてください。

キータイプ・ゲーム，完成っ！

これで本当に入力処理プログラムの完成です。ちょっとややこしいですが，ゲームの核を担うプログラムとしては非常に軽量に作られていると思います。この



方法なら，入力方法がすべてテーブル定義されているので，オリジナルな入力方法を生成してしまうことも簡単です。投稿プログラムや自主制作ゲームなどには十分用いることができると思いますので，どんどん使っちゃってください。市販ゲームで使われるときは…プロのプログラマーさんなら，もっと良いルーチンを考えてくださいな。

タイピング・ゲームはこの入力部分が核になるゲームですから，入力はできるだけ心地よくできるように，かつ，それだけじゃないおもしろさも付与して，総合的におもしろいゲームをぜひぜひ作ってくださいませ。

今回の話をはじめるときにも書きましたけれど，タイピング・ゲームはアクション性を多く含んでいながら，文章をユーザーに読ませる方法としても秀逸だと思っています。ゲーム機でもキーボードのオプションは当たり前になってきたこのごろですし，ぼくは，いろんなゲームが発表されることをひそかに期待しているのです。

《リスト1》 ローマ字処理のサンプル・プログラム・リスト

```
/*
** key type game sample
*/
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

// ローマ字テーブル
const unsigned char pszRingKeyTbl000[][8] = { "あ","A","" };
const unsigned char pszRingKeyTbl001[][8] = { "い","I","" };
const unsigned char pszRingKeyTbl002[][8] = { "う","U","" };
const unsigned char pszRingKeyTbl003[][8] = { "え","E","" };
const unsigned char pszRingKeyTbl004[][8] = { "お","O","" };
const unsigned char pszRingKeyTbl005[][8] = { "ぁ","LA","XA","" };
const unsigned char pszRingKeyTbl006[][8] = { "ぃ","LI","XI","" };
const unsigned char pszRingKeyTbl007[][8] = { "ぅ","LU","XU","" };
const unsigned char pszRingKeyTbl008[][8] = { "ぇ","LE","XE","" };
const unsigned char pszRingKeyTbl009[][8] = { "ぉ","LO","XO","" };
const unsigned char pszRingKeyTbl010[][8] = { "か","KA","" };
const unsigned char pszRingKeyTbl011[][8] = { "き","KI","" };
const unsigned char pszRingKeyTbl012[][8] = { "く","KU","" };
const unsigned char pszRingKeyTbl013[][8] = { "け","KE","" };
const unsigned char pszRingKeyTbl014[][8] = { "こ","KO","" };
const unsigned char pszRingKeyTbl015[][8] = { "きゃ","KYA","" };
const unsigned char pszRingKeyTbl016[][8] = { "きぃ","KYI","" };
const unsigned char pszRingKeyTbl017[][8] = { "きゅ","KYU","" };
const unsigned char pszRingKeyTbl018[][8] = { "きぇ","KYE","" };
const unsigned char pszRingKeyTbl019[][8] = { "きょ","KYO","" };
const unsigned char pszRingKeyTbl020[][8] = { "くぁ","QWA","KWA","" };
const unsigned char pszRingKeyTbl021[][8] = { "さ","SA","" };
const unsigned char pszRingKeyTbl022[][8] = { "し","SI","SHI","" };
const unsigned char pszRingKeyTbl023[][8] = { "す","SU","" };
const unsigned char pszRingKeyTbl024[][8] = { "せ","SE","" };
const unsigned char pszRingKeyTbl025[][8] = { "そ","SO","" };
const unsigned char pszRingKeyTbl026[][8] = { "しゃ","SYA","SHA","" };
const unsigned char pszRingKeyTbl027[][8] = { "しぃ","SYI","" };
const unsigned char pszRingKeyTbl028[][8] = { "しゅ","SYU","SHU","" };
const unsigned char pszRingKeyTbl029[][8] = { "しぇ","SYE","SHE","" };
const unsigned char pszRingKeyTbl030[][8] = { "しょ","SYO","SHO","" };
const unsigned char pszRingKeyTbl031[][8] = { "た","TA","" };
const unsigned char pszRingKeyTbl032[][8] = { "ち","TI","CHI","" };
const unsigned char pszRingKeyTbl033[][8] = { "つ","TU","TSU","" };
const unsigned char pszRingKeyTbl034[][8] = { "て","TE","" };
const unsigned char pszRingKeyTbl035[][8] = { "と","TO","" };
const unsigned char pszRingKeyTbl036[][8] = { "っ","LTU","XTU","" };
const unsigned char pszRingKeyTbl037[][8] = { "ちゃ","TYA","CYA","CHA","" };
const unsigned char pszRingKeyTbl038[][8] = { "ちぃ","TYI","CYI","" };
const unsigned char pszRingKeyTbl039[][8] = { "ちゅ","TYU","CYU","CHU","" };
const unsigned char pszRingKeyTbl040[][8] = { "ちぇ","TYE","CYE","CHE","" };
const unsigned char pszRingKeyTbl041[][8] = { "ちょ","TYO","CYO","CHO","" };
const unsigned char pszRingKeyTbl042[][8] = { "つぁ","TSA","" };
const unsigned char pszRingKeyTbl043[][8] = { "つぃ","TSI","" };
const unsigned char pszRingKeyTbl044[][8] = { "つぇ","TSE","" };
const unsigned char pszRingKeyTbl045[][8] = { "つぉ","TSO","" };
const unsigned char pszRingKeyTbl046[][8] = { "てゃ","THA","" };
const unsigned char pszRingKeyTbl047[][8] = { "てぃ","THI","" };
```

```
const unsigned char pszRingKeyTbl048[][8] = { "てゅ","THU","" };
const unsigned char pszRingKeyTbl049[][8] = { "てぇ","THE","" };
const unsigned char pszRingKeyTbl050[][8] = { "てょ","THO","" };
const unsigned char pszRingKeyTbl051[][8] = { "とぁ","TWA","" };
const unsigned char pszRingKeyTbl052[][8] = { "とぃ","TWI","" };
const unsigned char pszRingKeyTbl053[][8] = { "とぅ","TWU","" };
const unsigned char pszRingKeyTbl054[][8] = { "とぇ","TWE","" };
const unsigned char pszRingKeyTbl055[][8] = { "とぉ","TWO","" };
const unsigned char pszRingKeyTbl056[][8] = { "な","NA","" };
const unsigned char pszRingKeyTbl057[][8] = { "に","NI","" };
const unsigned char pszRingKeyTbl058[][8] = { "ぬ","NU","" };
const unsigned char pszRingKeyTbl059[][8] = { "ね","NE","" };
const unsigned char pszRingKeyTbl060[][8] = { "の","NO","" };
const unsigned char pszRingKeyTbl061[][8] = { "にゃ","NYA","" };
const unsigned char pszRingKeyTbl062[][8] = { "にぃ","NYI","" };
const unsigned char pszRingKeyTbl063[][8] = { "にゅ","NYU","" };
const unsigned char pszRingKeyTbl064[][8] = { "にぇ","NYE","" };
const unsigned char pszRingKeyTbl065[][8] = { "にょ","NYO","" };
const unsigned char pszRingKeyTbl066[][8] = { "は","HA","" };
const unsigned char pszRingKeyTbl067[][8] = { "ひ","HI","" };
const unsigned char pszRingKeyTbl068[][8] = { "ふ","HU","FU","" };
const unsigned char pszRingKeyTbl069[][8] = { "へ","HE","" };
const unsigned char pszRingKeyTbl070[][8] = { "ほ","HO","" };
const unsigned char pszRingKeyTbl071[][8] = { "ひゃ","HYA","" };
const unsigned char pszRingKeyTbl072[][8] = { "ひぃ","HYI","" };
const unsigned char pszRingKeyTbl073[][8] = { "ひゅ","HYU","" };
const unsigned char pszRingKeyTbl074[][8] = { "ひぇ","HYE","" };
const unsigned char pszRingKeyTbl075[][8] = { "ひょ","HYO","" };
const unsigned char pszRingKeyTbl076[][8] = { "ふぁ","FA","" };
const unsigned char pszRingKeyTbl077[][8] = { "ふぃ","FI","" };
const unsigned char pszRingKeyTbl078[][8] = { "ふぇ","FE","" };
const unsigned char pszRingKeyTbl079[][8] = { "ふぉ","FO","" };
const unsigned char pszRingKeyTbl080[][8] = { "ふゃ","FYA","" };
const unsigned char pszRingKeyTbl081[][8] = { "ふぃ","FYI","" };
const unsigned char pszRingKeyTbl082[][8] = { "ふゅ","FYU","" };
const unsigned char pszRingKeyTbl083[][8] = { "ふぇ","FYE","" };
const unsigned char pszRingKeyTbl084[][8] = { "ふょ","FYO","" };
const unsigned char pszRingKeyTbl085[][8] = { "ま","MA","" };
const unsigned char pszRingKeyTbl086[][8] = { "み","MI","" };
const unsigned char pszRingKeyTbl087[][8] = { "む","MU","" };
const unsigned char pszRingKeyTbl088[][8] = { "め","ME","" };
const unsigned char pszRingKeyTbl089[][8] = { "も","MO","" };
const unsigned char pszRingKeyTbl090[][8] = { "みゃ","MYA","" };
const unsigned char pszRingKeyTbl091[][8] = { "みぃ","MYI","" };
const unsigned char pszRingKeyTbl092[][8] = { "みゅ","MYU","" };
const unsigned char pszRingKeyTbl093[][8] = { "みぇ","MYE","" };
const unsigned char pszRingKeyTbl094[][8] = { "みょ","MYO","" };
const unsigned char pszRingKeyTbl095[][8] = { "や","YA","" };
const unsigned char pszRingKeyTbl096[][8] = { "い","YI","" };
const unsigned char pszRingKeyTbl097[][8] = { "ゆ","YU","" };
const unsigned char pszRingKeyTbl098[][8] = { "いぇ","YE","" };
const unsigned char pszRingKeyTbl099[][8] = { "よ","YO","" };
const unsigned char pszRingKeyTbl100[][8] = { "ゃ","LYA","XYA","" };
const unsigned char pszRingKeyTbl101[][8] = { "ぃ","LYI","XYI","" };
const unsigned char pszRingKeyTbl102[][8] = { "ゅ","LYU","XYU","" };
const unsigned char pszRingKeyTbl103[][8] = { "ぇ","LYE","XYE","" };
const unsigned char pszRingKeyTbl104[][8] = { "ょ","LYO","XYO","" };
const unsigned char pszRingKeyTbl105[][8] = { "ら","RA","" };
const unsigned char pszRingKeyTbl106[][8] = { "り","RI","" };
const unsigned char pszRingKeyTbl107[][8] = { "る","RU","" };
const unsigned char pszRingKeyTbl108[][8] = { "れ","RE","" };
const unsigned char pszRingKeyTbl109[][8] = { "ろ","RO","" };
const unsigned char pszRingKeyTbl110[][8] = { "りゃ","RYA","" };
const unsigned char pszRingKeyTbl111[][8] = { "りぃ","RYI","" };
const unsigned char pszRingKeyTbl112[][8] = { "りゅ","RYU","" };
const unsigned char pszRingKeyTbl113[][8] = { "りぇ","RYE","" };
const unsigned char pszRingKeyTbl114[][8] = { "りょ","RYO","" };
const unsigned char pszRingKeyTbl115[][8] = { "わ","WA","" };
const unsigned char pszRingKeyTbl116[][8] = { "うぃ","WI","" };
const unsigned char pszRingKeyTbl117[][8] = { "う","WU","" };
const unsigned char pszRingKeyTbl118[][8] = { "うぇ","WE","" };
const unsigned char pszRingKeyTbl119[][8] = { "を","WO","" };
const unsigned char pszRingKeyTbl120[][8] = { "ん","N","" };
const unsigned char pszRingKeyTbl121[][8] = { "が","GA","" };
const unsigned char pszRingKeyTbl122[][8] = { "ぎ","GI","" };
const unsigned char pszRingKeyTbl123[][8] = { "ぐ","GU","" };
const unsigned char pszRingKeyTbl124[][8] = { "げ","GE","" };
const unsigned char pszRingKeyTbl125[][8] = { "ご","GO","" };
const unsigned char pszRingKeyTbl126[][8] = { "ぎゃ","GYA","" };
const unsigned char pszRingKeyTbl127[][8] = { "ぎぃ","GYI","" };
const unsigned char pszRingKeyTbl128[][8] = { "ぎゅ","GYU","" };
const unsigned char pszRingKeyTbl129[][8] = { "ぎぇ","GYE","" };
const unsigned char pszRingKeyTbl130[][8] = { "ぎょ","GYO","" };
const unsigned char pszRingKeyTbl131[][8] = { "ぐぁ","GWA","" };
const unsigned char pszRingKeyTbl132[][8] = { "ぐぃ","GWI","" };
const unsigned char pszRingKeyTbl133[][8] = { "ぐぅ","GWU","" };
const unsigned char pszRingKeyTbl134[][8] = { "ぐぇ","GWE","" };
const unsigned char pszRingKeyTbl135[][8] = { "ぐぉ","GWO","" };
const unsigned char pszRingKeyTbl136[][8] = { "ざ","ZA","" };
const unsigned char pszRingKeyTbl137[][8] = { "じ","ZI","JI","" };
const unsigned char pszRingKeyTbl138[][8] = { "ず","ZU","" };
const unsigned char pszRingKeyTbl139[][8] = { "ぜ","ZE","" };
const unsigned char pszRingKeyTbl140[][8] = { "ぞ","ZO","" };
const unsigned char pszRingKeyTbl141[][8] = { "じゃ","JYA","ZYA","JA","" };
const unsigned char pszRingKeyTbl142[][8] = { "じぃ","JYI","ZYI","" };
const unsigned char pszRingKeyTbl143[][8] = { "じゅ","JYU","ZYU","JU","" };
const unsigned char pszRingKeyTbl144[][8] = { "じぇ","JYE","ZYE","JE","" };
const unsigned char pszRingKeyTbl145[][8] = { "じょ","JYO","ZYO","JO","" };
const unsigned char pszRingKeyTbl146[][8] = { "だ","DA","" };
const unsigned char pszRingKeyTbl147[][8] = { "ぢ","DI","" };
const unsigned char pszRingKeyTbl148[][8] = { "づ","DU","" };
const unsigned char pszRingKeyTbl149[][8] = { "で","DE","" };
const unsigned char pszRingKeyTbl150[][8] = { "ど","DO","" };
const unsigned char pszRingKeyTbl151[][8] = { "ぢゃ","DYA","" };
const unsigned char pszRingKeyTbl152[][8] = { "ぢぃ","DYI","" };
const unsigned char pszRingKeyTbl153[][8] = { "ぢゅ","DYU","" };
const unsigned char pszRingKeyTbl154[][8] = { "ぢぇ","DYE","" };
const unsigned char pszRingKeyTbl155[][8] = { "ぢょ","DYO","" };
const unsigned char pszRingKeyTbl156[][8] = { "でゃ","DHA","" };
const unsigned char pszRingKeyTbl157[][8] = { "でぃ","DHI","" };
const unsigned char pszRingKeyTbl158[][8] = { "でゅ","DHU","" };
const unsigned char pszRingKeyTbl159[][8] = { "でぇ","DHE","" };
const unsigned char pszRingKeyTbl160[][8] = { "でょ","DHO","" };
const unsigned char pszRingKeyTbl161[][8] = { "どぁ","DWA","" };
const unsigned char pszRingKeyTbl162[][8] = { "どぃ","DWI","" };
const unsigned char pszRingKeyTbl163[][8] = { "どぅ","DWU","" };
const unsigned char pszRingKeyTbl164[][8] = { "どぇ","DWE","" };
const unsigned char pszRingKeyTbl165[][8] = { "どぉ","DWO","" };
const unsigned char pszRingKeyTbl166[][8] = { "ば","BA","" };
const unsigned char pszRingKeyTbl167[][8] = { "び","BI","" };
const unsigned char pszRingKeyTbl168[][8] = { "ぶ","BU","" };
const unsigned char pszRingKeyTbl169[][8] = { "べ","BE","" };
const unsigned char pszRingKeyTbl170[][8] = { "ぼ","BO","" };
const unsigned char pszRingKeyTbl171[][8] = { "びゃ","BYA","" };
const unsigned char pszRingKeyTbl172[][8] = { "びぃ","BYI","" };
const unsigned char pszRingKeyTbl173[][8] = { "びゅ","BYU","" };
const unsigned char pszRingKeyTbl174[][8] = { "びぇ","BYE","" };
const unsigned char pszRingKeyTbl175[][8] = { "びょ","BYO","" };
const unsigned char pszRingKeyTbl176[][8] = { "ぱ","PA","" };
const unsigned char pszRingKeyTbl177[][8] = { "ぴ","PI","" };
const unsigned char pszRingKeyTbl178[][8] = { "ぷ","PU","" };
const unsigned char pszRingKeyTbl179[][8] = { "ぺ","PE","" };
const unsigned char pszRingKeyTbl180[][8] = { "ぽ","PO","" };
const unsigned char pszRingKeyTbl181[][8] = { "ぴゃ","PYA","" };
const unsigned char pszRingKeyTbl182[][8] = { "ぴぃ","PYI","" };
const unsigned char pszRingKeyTbl183[][8] = { "ぴゅ","PYU","" };
const unsigned char pszRingKeyTbl184[][8] = { "ぴぇ","PYE","" };
const unsigned char pszRingKeyTbl185[][8] = { "ぴょ","PYO","" };
const unsigned char pszRingKeyTbl186[][8] = { "ヴぁ","VA","" };
const unsigned char pszRingKeyTbl187[][8] = { "ヴぃ","VI","" };
const unsigned char pszRingKeyTbl188[][8] = { "ヴ","VU","" };
const unsigned char pszRingKeyTbl189[][8] = { "ヴぇ","VE","" };
const unsigned char pszRingKeyTbl190[][8] = { "ヴぉ","VO","" };
const unsigned char pszRingKeyTbl191[][8] = { "ヴゃ","VYA","" };
const unsigned char pszRingKeyTbl192[][8] = { "ヴぃ","VYI","" };
const unsigned char pszRingKeyTbl193[][8] = { "ヴゅ","VYU","" };
const unsigned char pszRingKeyTbl194[][8] = { "ヴぇ","VYE","" };
const unsigned char pszRingKeyTbl195[][8] = { "ー","-","" };


char *pRingKeyTbl[] = {

(char*)pszRingKeyTbl000,(char*)pszRingKeyTbl001,(char*)pszRingKeyTbl002,(cha
r*)pszRingKeyTbl003,(char*)pszRingKeyTbl004,

(char*)pszRingKeyTbl005,(char*)pszRingKeyTbl006,(char*)pszRingKeyTbl007,(cha
r*)pszRingKeyTbl008,(char*)pszRingKeyTbl009,

(char*)pszRingKeyTbl010,(char*)pszRingKeyTbl011,(char*)pszRingKeyTbl012,(cha
r*)pszRingKeyTbl013,(char*)pszRingKeyTbl014,

(char*)pszRingKeyTbl015,(char*)pszRingKeyTbl016,(char*)pszRingKeyTbl017,(cha
r*)pszRingKeyTbl018,(char*)pszRingKeyTbl019,

(char*)pszRingKeyTbl020,(char*)pszRingKeyTbl021,(char*)pszRingKeyTbl022,(cha
r*)pszRingKeyTbl023,(char*)pszRingKeyTbl024,

(char*)pszRingKeyTbl025,(char*)pszRingKeyTbl026,(char*)pszRingKeyTbl027,(cha
r*)pszRingKeyTbl028,(char*)pszRingKeyTbl029,

(char*)pszRingKeyTbl030,(char*)pszRingKeyTbl031,(char*)pszRingKeyTbl032,(cha
r*)pszRingKeyTbl033,(char*)pszRingKeyTbl034,

(char*)pszRingKeyTbl035,(char*)pszRingKeyTbl036,(char*)pszRingKeyTbl037,(cha
r*)pszRingKeyTbl038,(char*)pszRingKeyTbl039,

(char*)pszRingKeyTbl040,(char*)pszRingKeyTbl041,(char*)pszRingKeyTbl042,(cha
r*)pszRingKeyTbl043,(char*)pszRingKeyTbl044,

(char*)pszRingKeyTbl045,(char*)pszRingKeyTbl046,(char*)pszRingKeyTbl047,(cha
r*)pszRingKeyTbl048,(char*)pszRingKeyTbl049,

(char*)pszRingKeyTbl050,(char*)pszRingKeyTbl051,(char*)pszRingKeyTbl052,(cha
r*)pszRingKeyTbl053,(char*)pszRingKeyTbl054,

(char*)pszRingKeyTbl055,(char*)pszRingKeyTbl056,(char*)pszRingKeyTbl057,(cha
r*)pszRingKeyTbl058,(char*)pszRingKeyTbl059,

(char*)pszRingKeyTbl060,(char*)pszRingKeyTbl061,(char*)pszRingKeyTbl062,(cha
r*)pszRingKeyTbl063,(char*)pszRingKeyTbl064,

(char*)pszRingKeyTbl065,(char*)pszRingKeyTbl066,(char*)pszRingKeyTbl067,(cha
r*)pszRingKeyTbl068,(char*)pszRingKeyTbl069,

(char*)pszRingKeyTbl070,(char*)pszRingKeyTbl071,(char*)pszRingKeyTbl072,(cha
r*)pszRingKeyTbl073,(char*)pszRingKeyTbl074,

(char*)pszRingKeyTbl075,(char*)pszRingKeyTbl076,(char*)pszRingKeyTbl077,(cha
r*)pszRingKeyTbl078,(char*)pszRingKeyTbl079,

(char*)pszRingKeyTbl080,(char*)pszRingKeyTbl081,(char*)pszRingKeyTbl082,(cha
r*)pszRingKeyTbl083,(char*)pszRingKeyTbl084,

(char*)pszRingKeyTbl085,(char*)pszRingKeyTbl086,(char*)pszRingKeyTbl087,(cha
r*)pszRingKeyTbl088,(char*)pszRingKeyTbl089,

(char*)pszRingKeyTbl090,(char*)pszRingKeyTbl091,(char*)pszRingKeyTbl092,(cha
r*)pszRingKeyTbl093,(char*)pszRingKeyTbl094,

(char*)pszRingKeyTbl095,(char*)pszRingKeyTbl096,(char*)pszRingKeyTbl097,(cha
r*)pszRingKeyTbl098,(char*)pszRingKeyTbl099,

(char*)pszRingKeyTbl100,(char*)pszRingKeyTbl101,(char*)pszRingKeyTbl102,(cha
r*)pszRingKeyTbl103,(char*)pszRingKeyTbl104,

(char*)pszRingKeyTbl105,(char*)pszRingKeyTbl106,(char*)pszRingKeyTbl107,(cha
r*)pszRingKeyTbl108,(char*)pszRingKeyTbl109,

(char*)pszRingKeyTbl110,(char*)pszRingKeyTbl111,(char*)pszRingKeyTbl112,(cha
r*)pszRingKeyTbl113,(char*)pszRingKeyTbl114,
```

```
(char*)pszRingKeyTbl115,(char*)pszRingKeyTbl116,(char*)pszRingKeyTbl117,(cha
r*)pszRingKeyTbl118,(char*)pszRingKeyTbl119,

(char*)pszRingKeyTbl120,(char*)pszRingKeyTbl121,(char*)pszRingKeyTbl122,(cha
r*)pszRingKeyTbl123,(char*)pszRingKeyTbl124,

(char*)pszRingKeyTbl125,(char*)pszRingKeyTbl126,(char*)pszRingKeyTbl127,(cha
r*)pszRingKeyTbl128,(char*)pszRingKeyTbl129,

(char*)pszRingKeyTbl130,(char*)pszRingKeyTbl131,(char*)pszRingKeyTbl132,(cha
r*)pszRingKeyTbl133,(char*)pszRingKeyTbl134,

(char*)pszRingKeyTbl135,(char*)pszRingKeyTbl136,(char*)pszRingKeyTbl137,(cha
r*)pszRingKeyTbl138,(char*)pszRingKeyTbl139,

(char*)pszRingKeyTbl140,(char*)pszRingKeyTbl141,(char*)pszRingKeyTbl142,(cha
r*)pszRingKeyTbl143,(char*)pszRingKeyTbl144,

(char*)pszRingKeyTbl145,(char*)pszRingKeyTbl146,(char*)pszRingKeyTbl147,(cha
r*)pszRingKeyTbl148,(char*)pszRingKeyTbl149,

(char*)pszRingKeyTbl150,(char*)pszRingKeyTbl151,(char*)pszRingKeyTbl152,(cha
r*)pszRingKeyTbl153,(char*)pszRingKeyTbl154,

(char*)pszRingKeyTbl155,(char*)pszRingKeyTbl156,(char*)pszRingKeyTbl157,(cha
r*)pszRingKeyTbl158,(char*)pszRingKeyTbl159,

(char*)pszRingKeyTbl160,(char*)pszRingKeyTbl161,(char*)pszRingKeyTbl162,(cha
r*)pszRingKeyTbl163,(char*)pszRingKeyTbl164,

(char*)pszRingKeyTbl165,(char*)pszRingKeyTbl166,(char*)pszRingKeyTbl167,(cha
r*)pszRingKeyTbl168,(char*)pszRingKeyTbl169,

(char*)pszRingKeyTbl170,(char*)pszRingKeyTbl171,(char*)pszRingKeyTbl172,(cha
r*)pszRingKeyTbl173,(char*)pszRingKeyTbl174,

(char*)pszRingKeyTbl175,(char*)pszRingKeyTbl176,(char*)pszRingKeyTbl177,(cha
r*)pszRingKeyTbl178,(char*)pszRingKeyTbl179,

(char*)pszRingKeyTbl180,(char*)pszRingKeyTbl181,(char*)pszRingKeyTbl182,(cha
r*)pszRingKeyTbl183,(char*)pszRingKeyTbl184,

(char*)pszRingKeyTbl185,(char*)pszRingKeyTbl186,(char*)pszRingKeyTbl187,(cha
r*)pszRingKeyTbl188,(char*)pszRingKeyTbl189,

(char*)pszRingKeyTbl190,(char*)pszRingKeyTbl191,(char*)pszRingKeyTbl192,(cha
r*)pszRingKeyTbl193,(char*)pszRingKeyTbl194,
  (char*)pszRingKeyTbl195, 0
};

// p1とp2を比べるサブルーチン。p1の長さに合わせる
int StringCheck( char *p1, char *p2 )
{
  int i;
  for ( i = 0; i < (int)strlen( p1 ); i++ ) {
    if ( p1[i] != p2[i] ) { return 0; }
  }
  return 1;
}

// メインプログラムはここから。
void main()
{
  char szQuestion[] = "きゃっとらいくのきーたいぷぷろぐらむ";
  int nMakeTblFlg;  // テーブル作成フラグ
  char szWordTbl[64][8]; // 入力させる1ワード保存テーブル
  char szKeyTbl[64][8]; // 入力させる1ワード対応キー保存テーブル
  int nUseTblCnt;     // nUseTblのカウント数
  int nInCnt;         // 問題読み進みカウンタ
  int nKeyCnt;     // 入力読み進みカウンタ
  int nDelTblMark[64];  // 削除予約マーク
  int nSuccessFlg;  // 入力成功フラグ
  int nEndFlg;     // 入力終了フラグ
  int nViewFlg;      // 入力文字表示フラグ
  int nSpecial_N_Flg;   // 特例。「ん」の「N」及び「NN」入力用

  int i, j; char *p, cKey; // 汎用

  // とりあえず入力内容表示
  printf( "「%s」を入力せよ！\n", szQuestion );

  // 変数初期化
  nEndFlg = 0;
  nSpecial_N_Flg = 0;
  nInCnt = 0;
  nMakeTblFlg = 1;
  // 入力メインループ
  while( nEndFlg == 0 ) {
    // 入力テーブルを作成する必要があるか？
    if ( nMakeTblFlg == 1 ) {
      //--------------- ここから、入力テーブル作成
      // ローマ字テーブルから、今回の入力時(1ワード分)に使う入力テーブルを抽出
      i = 0; nUseTblCnt = 0; nKeyCnt = 0;
      while( ( p = pRingKeyTbl[i] ) != 0 ) {
        if ( StringCheck( &p[ 0 * 8 ], &szQuestion[ nInCnt ] ) == 1 ) {
          // 次の入力文字とローマ字テーブルが一致した
          // szWordTblにテーブル構築
          j = 1;
          while( p[ j * 8 ] != '\0' ) {
            strcpy( szWordTbl[nUseTblCnt], &p[ 0 * 8 ] );
            strcpy( szKeyTbl[nUseTblCnt], &p[ j * 8 ] );
            j++; nUseTblCnt++;
          }
        }
        i++; // 次のテーブルを参照しにいく
      }
      // 「っ」特例判定
      if ( StringCheck( "っ", &szQuestion[ nInCnt ] ) == 1 ) {
        // 「っ」特例処理を行う
        i = 0;
        while( ( p = pRingKeyTbl[i] ) != 0 ) {
          if ( StringCheck( &p[ 0 * 8 ], &szQuestion[ nInCnt + 2 ] ) == 1 ) {
            // 次の入力文字とローマ字テーブルが一致した
            // szWordTblにテーブル構築
            j = 1;
            while( p[ j * 8 ] != '\0' ) {
              // 「っ」を頭につける
              strcpy( szWordTbl[nUseTblCnt], "っ" );
              strcat( szWordTbl[nUseTblCnt], &p[ 0 * 8 ] );
              // 次の音を重ねる（ただし、次は必ず子音とする。「っあ」等非対応）
              szKeyTbl[nUseTblCnt][0] = p[ j * 8 ];
              strcpy( &szKeyTbl[nUseTblCnt][1], &p[ j * 8 ] );
              j++; nUseTblCnt++;
            }
          }
          i++; // 次のテーブルを参照しにいく
        }
      }
      nMakeTblFlg = 0;
      //--------------- ここまで、入力テーブル作成
    }

    // 1キー分、入力されるまで待つ
    cKey = getch();

    cKey = toupper( cKey );        // 入力された文字を大文字化(判定単純化用)
    nSuccessFlg = 0;     // 入力成功フラグの初期化
    nViewFlg = 0;       // 入力文字表示フラグの初期化

    // 入力された文字は、今回の入力に使える文字か？
    for ( i = 0; i < nUseTblCnt; i++ ) {
      // まだ有効なテーブルか？
      if ( szWordTbl[ i ][0] != '\0' ) {
        // 有効なテーブルである
        // ローマ字テーブルと照らし合わせる
        if ( szKeyTbl[ i ][ nKeyCnt ] == cKey ) {
          // 入力成功
          // 成功フラグを立てる。この入力は消さないので削除予約マークはつけない
          nSuccessFlg = 1;
          nDelTblMark[ i ] = 0;
        } else {
          // このテーブル内容と入力文字は一致しない
          // この入力に削除予約マークをつける
          nDelTblMark[ i ] = 1;
        }
      }
    } // of for. 入力テーブル分

    // 入力成功しているか？
    if ( nSuccessFlg == 1 ) {
      // 入力成功している
      nSpecial_N_Flg = 0; // 「ん」特例フラグの初期化
      nSuccessFlg = 0;// 今度は1ワード入力終了判定に使う
      nKeyCnt++;     // 次の入力キー場所に
      for ( i = 0; i < nUseTblCnt; i++ ) {
        // まだ有効なテーブルか？
        if ( szWordTbl[ i ][0] != '\0' ) {
          // 有効なテーブルである
          if ( nDelTblMark[ i ] == 1 ) {
            // このテーブルを次から無効にする
            szWordTbl[ i ][0] = '\0';
          } else {
            // このテーブルはまだ有効
            // 1ワード入力終了したか？
            if ( szKeyTbl[ i ][ nKeyCnt ] == '\0' ) {
              // 1ワード入力終了
              nMakeTblFlg = 1;                 // 新たな入力テーブルを作る
              // 問題読み込みカウンタを進める
              nInCnt += strlen( szWordTbl[ i ] );
              // 問題は打ち終わったか
              if ( szQuestion[ nInCnt ] == 0 ) {
                // 打ち終わったのでループから出る
                nEndFlg = 1;
              } else {
                // 打ち終わってない
                if ( strcmp( szWordTbl[ i ], "ん" ) == 0 ) {
                  // 「ん」特例。次の N はミスとカウントしない
                  nSpecial_N_Flg = 1;
                }
              }
            }
          }
        }
      } // of for. 入力テーブル分
      nViewFlg = 1;       // 入力文字を表示する
    } else {
      // 入力失敗
      if ( nSpecial_N_Flg == 1 && cKey == 'N' ) {
        // 「ん」特例。今回はミスとカウントしない
        nViewFlg = 1;       // 入力文字を表示する
      } else {
        // ゲームによってはダメージ判定とかだね
        printf( "!!error!!" );
      }
      nSpecial_N_Flg = 0; // 「ん」特例フラグの初期化
    }

    // 入力文字の表示
    if ( nViewFlg == 1 ) {
      printf( "%c", cKey );
    }

  } // of while. メインループ

  printf( "\n" );

}
```

パソコンでロボットを操る！

レゴ道場 桐林部屋

チャレンジ!! レゴ・マインドストーム

LEGO MINDSTORMS ROBOTICS INVENTION SYSTEM



お題：センサーで自動ドアを作る

こんにちは，桐林です。今回は，いよいよセンサーを使います。自動ドアを作ってみましょう!!!　前回の問題の答えもあります。　（文/桐林星河）

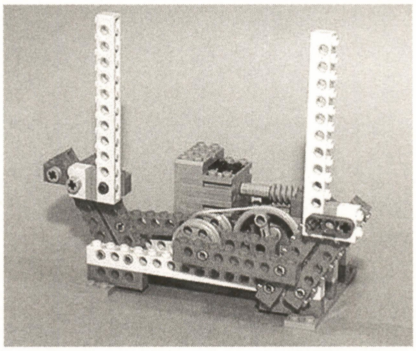


『LEGO MINDSTORMS Robotics Invention System 2.0（日本語版）』
実売価格：約23,800円
対応OS：Windows98/Me
総パーツ数：718（レゴブロック/RCX/USB IRタワー/モータ2個/タッチセンサ2個/ライトセンサ1個/日本語マニュアル/日本語ソフトウエアなど）

まずは先月号の宿題の答え

前回の作例「バスのワイパー」の「左右のワイパーの動きを同時にする」という問題。答えは簡単なものです。**写真1**を見てください。すべてのギヤの中心軸を，ワイパーについている連結棒の接続位置と同じ高さにすればいいのです。しかし，そうすると稼動範囲が狭くなってしまうので，ギヤをプーリーに交換して，稼動範囲を広くしています。ワイパーの付け根も改造してあります。

どうでしたか？ 予想どおりの答えでしたか？ これ以外の答えを考えた人はぜひ教えてください。ハガキかメール（webmas@basicmagazine.net）でお待ちしています。



《写真1》先月号のワイパーの答えです。わかりにくいですが，プーリーの回転軸と連結棒の軸の高さが同じになっています

いろいろなセンサーを理解しよう

タッチセンサー

タッチセンサーとは，RISやRDSに入ってくるセンサーの一つです（**写真2**）。タッチセンサーはいろいろなものに使えます。一般的には，壁に触ったかどうかを調べたり，タッチセンサーでリモコンを作ったりなどなど。タッチセンサーは名前のとおり，物に触ったら反応するわけです。これを応用して物を作っていきます。

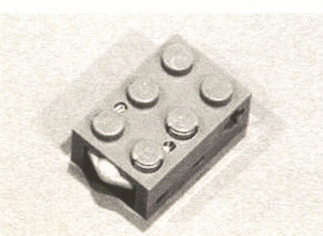
たとえば，回転する軸に突起を取り付けて，その突起に当たるようにタッチセンサーをセットすれば，回転数がわかる回転センサーになります（**写真3，4**）。また，リモコンと一口に言っても，いろいろな形のものが工夫しだいで作ることができます。ロボットアームのリモコンもその例で，ほとんどがタッチセンサーで制御されています（**写真5**）。

RISの説明書に載っているタッチセンサーの使いかたは，車などにつけて，壁などの障害物がないかチェックするものです。つまり，動くものにセンサーをつけて，動かないものをチェックしています。当然この逆もできるわけです。先ほどふれた回転センサーもその一例です。そのほかにも，今回挑戦する自動ドアのような例もあります。

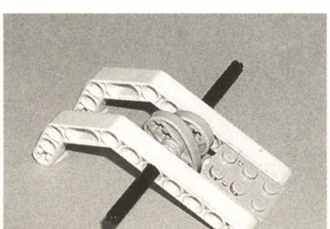
ライトセンサー

ライトセンサーもRISに入ってきます（**写真6**）。ライトセンサーは明るさを調べるためのセンサーです。RISの説明書では，黒いラインの上を走る車の作例が出ていますよね。これは，黒のほうが光を吸収して暗く，白のほうが光を反射して明るくなることを利用しています。なので，黄色と青など明るさのちがいが大きいものは感知できます。それにもちろん普通に「暗い」「明るい」もわかるので，通過検知器なども作れます。

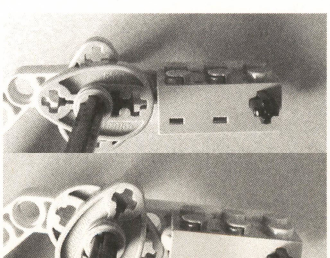
ライトセンサーは使うときには，いろ



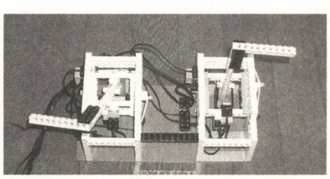
《写真2》これがタッチセンサー。RISに入っている



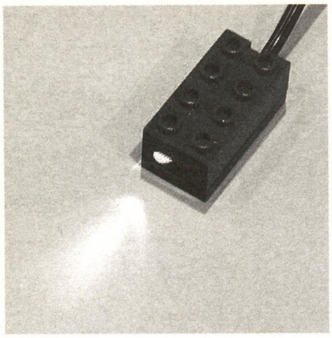
《写真3》タッチセンサーで作った回転センサーの例



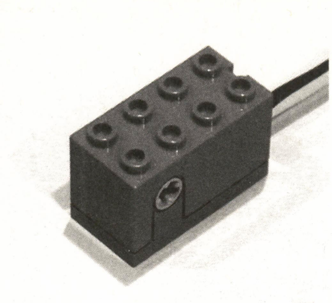
《写真4》上が押されている状態，下が押されていない状態



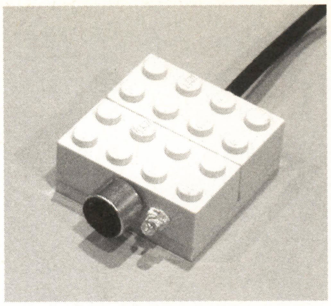
《写真5》ロボットアームのリモコン。前に倒したり，手前に倒したりして，アームを動かすのに使う



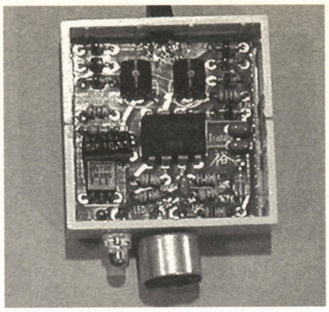
《写真6》ライトセンサー。これもRISに入っている



《写真7》これは回転センサー。バラ売りされている



《写真8》音量センサー。ブロックを削って，中に基板を入れる。トライステートで購入



《写真9》音量センサーの裏側。基板が剥き出しなので，普段は4×4のプレートをつけている

いろと注意しないといけないことがあります。たとえばセンサーの個体差が大きいということや，回りの明るさが変わるとうまく動かないことなどです。そこで，こまめにその場の明るさをチェックして，その値をプログラムに組み込んでいきます。みなさんも，説明書に載っていないような，おもしろくて，すごい使いかたを考えてみてください。

その他のセンサー

RISに入っているセンサー以外にも，ローテーションセンサー(回転センサー)，温度センサーがあります(**写真7**)。この二つはバラ売りでいろいろなお店で売られています。

回転センサーは，この名のとおり，センサーの穴にシャフトを通してシャフトの回転数を調べるというものです。これで移動した距離などがわかりますし，どれぐらい動かしたかなどということもわかります。

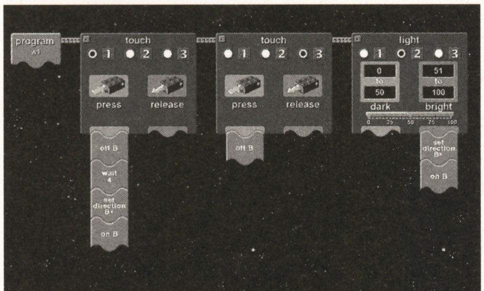
温度センサーは，もちろん温度を測るものです。でもレゴでは使い道があまりないかもしれないです(僕は使ったことがない)。

そのほかにも，音量センサーというのがあります(**写真8，9**)。レゴ社純正ではないのですが，トライステート(http://www.tristate.ne.jp)という会社から販売されています。これは音を感知するセンサーです。音のレベルはライトセンサーモードを使って0から100の値となって出てきます。このセンサーは，白川さんという方が開発したものです(http://www.geocities.co.jp/Technopolis/6264/mindstorms/index.html)。白川さんは，MINDSTORMS販売当初からのユーザーで，「目玉の親父」や「ゴンスケ」というおもしろい作品を作っていましたが，現在は青年海外協力隊員としてニジェールで活躍されています。

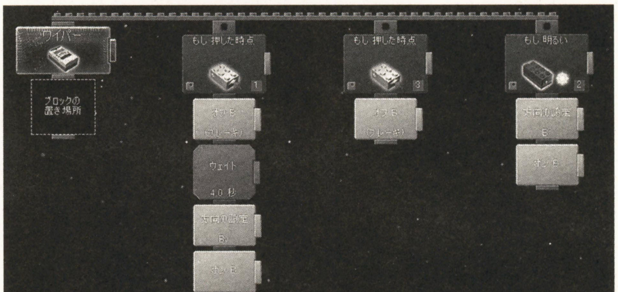
いよいよ「自動ドア」を製作

センサーの種類はわかりましたか？では今回のお題である，タッチセンサーとライトセンサーを組み合わせて「自動ドア」を組み立ててみましょう(**写真10**)。ドアの前に物が来ると，ドアが開いて，何秒かたつと閉まります。実際の自動ドアというのは，赤外線で物体を見つけていますが，今回作るものは可視光で判別しています。ですので，いまいち性能はよくないです。あしからず。とは言っても，実際に物体が来て，ドアが開くと感動しますよ。ぜひ挑戦してみてください。今回ももちろんRISのセットだけで作れます。プログラムは**第1図**と**第2図**を参考にしてください。

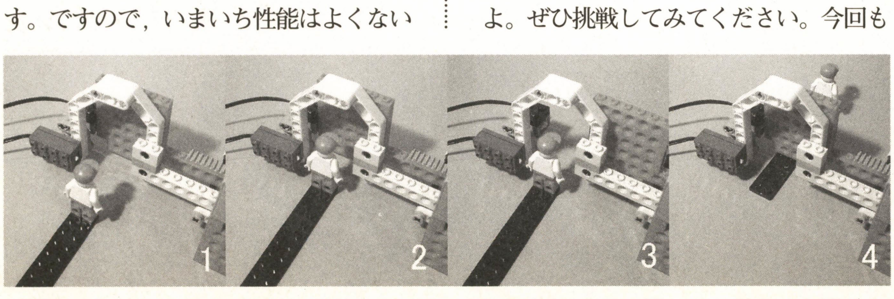
タッチセンサーとライトセンサーを組み合わせた「自動ドア」のプログラム



《第1図》RIS1.5でのプログラム



《第2図》RIS2.0でのプログラム

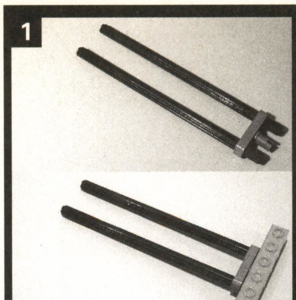


《写真10》左から，人が近づいてきて，センサーの前に立ち，ドアが開いて，4秒たつと閉まる

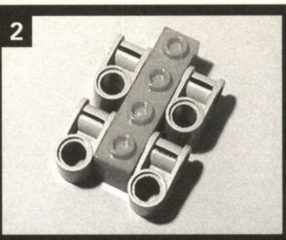
部品一覧

1×16のビーム	2	シャフトペグ	2
12のシャフト	2	1×3L型ハーフビーム	2
1×12のビーム	2	3／4ペグ	5
8のシャフト	1	1×3のハーフビーム	1
1×8のビーム	1	1／2ペグ	2
6のシャフト	1	1×2のプレート	2
1×6のビーム	6	ペグ	2
4のシャフト	1	24枚歯ギヤ	2
1×4のビーム	9	クロスブロック	4
3のシャフト	2	8枚歯ギヤ	1
1×2のビーム	4	タッチセンサー	2
2のシャフト	1	ラックギヤ	2
2角のアングルビーム	2	ライトセンサー	1
ハーフブッシュ	7	2×4のブロック	1
1×5のハーフビーム	2	モーター	1

1

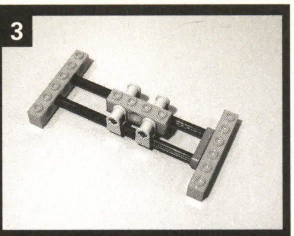


2



▲スライドする部分です。クロスブロックに3のシャフトを固定して作ります。

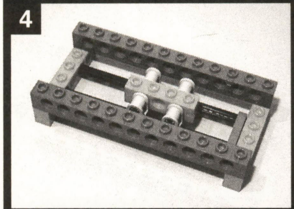
3



▲それを1で作ったスライド機構の土台につけます。このとき同時に，1×6のビームをさしておきます。

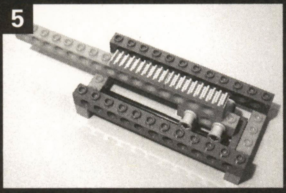
◀スライド機構を作ります。最初に上の写真のように12のシャフトに，ハーフビームを固定し，つぎに下の写真のように3/4ペグで1×6のビームに固定します。

4



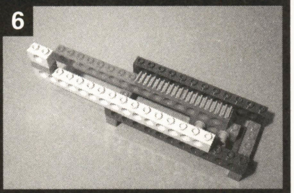
▲1×12のビームで固定します。

5



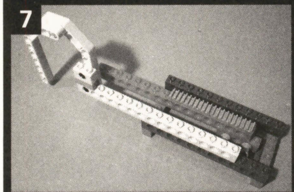
▲スライドする部分に1×16のビームを取り付けます。同時に1×16のビームにラックギヤを取り付けておきます。

6



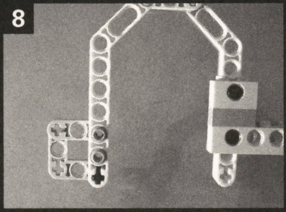
▲ドア枠の固定用のビームを取り付けます。先端には1×2プレート2枚と1×2のブロックを取り付けます。

7



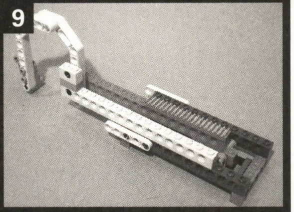
▲ドア枠を取り付けます。

8



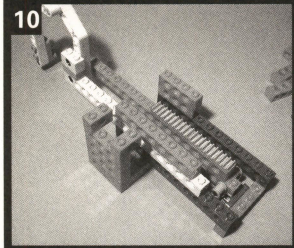
▲センサーの土台をドア枠につけます。1/2ペグも忘れないでください。

9



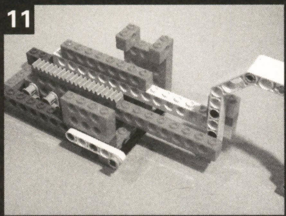
▲1×5のハーフビームを左右につけます。

10



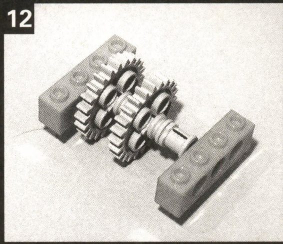
▲モーターの土台をつけます。ブロックの数に注意してください。

11



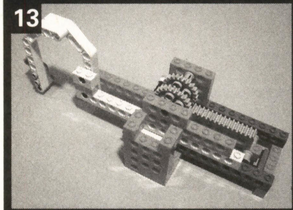
▲9でつけたハーフビームに8のシャフトを通して，ハーフブッシュで固定します。

12



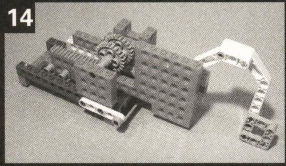
▲ギヤを作ります。

13



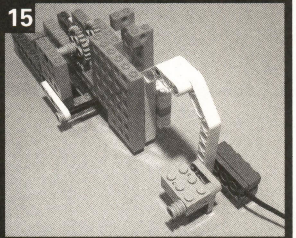
▲ギヤを土台に取り付けます。

14



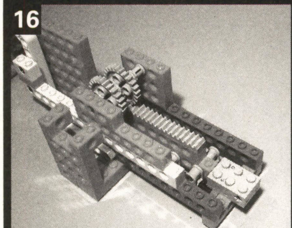
▲ドアをつけます。

15



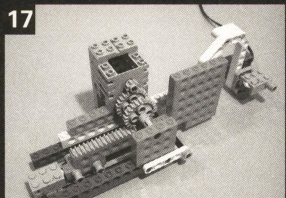
▲センサーの取り付けです。写真はドア枠付近。

16



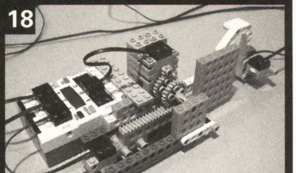
▲センサーの取り付けです。写真はドア機械部分。

17



▲モーターを取り付けます。

18



▲配線をして完成です。配線はモーターBにドア開閉用モーター，センサー1にドア機械部タッチセンサー，センサー2にライトセンサー，センサー3にドア枠のタッチセンサーです。

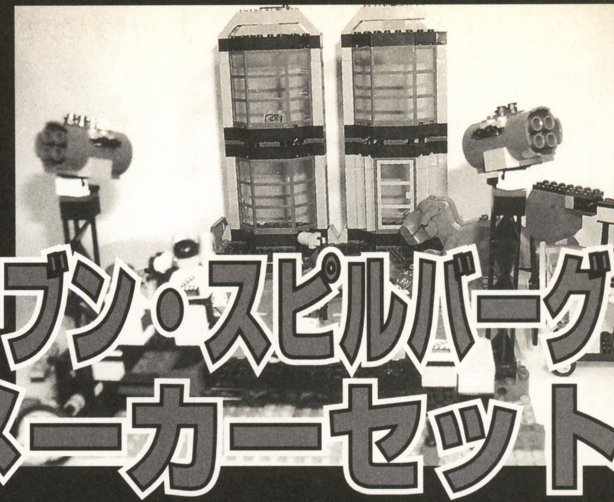
次回はライトセンサーを使って，ライントレーサーを作ってみましょう。速さを競う競技なんかもできる，じつは奥の深いロボットです。それでは!!

桐林星河（きりばやしせいが）
機械やロボットが好きな14歳。小学生の頃からメカに慣れ親しみ，持ち前の努力と根性で，LEGO MINDSTORMSの競技などでは常に好成績。TV出演なども多数。本誌ではライターとして活躍中だが，中学3年の受験生。

レゴで映画が撮れる!!

レゴスタジオ LEGO STUDIOS

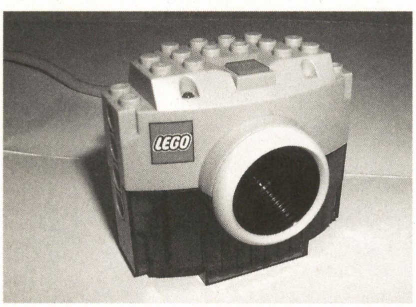




レゴ&スティーブン・スピルバーグ ムービーメーカーセット

レゴスタジオとは？

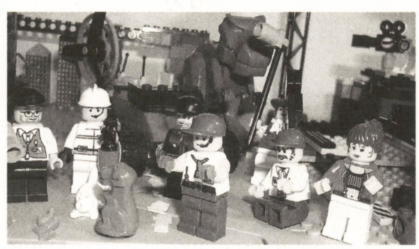
今回新たに発売された「レゴスタジオ(LEGO Studios)」。これはなんと“パソコンでレゴの映画が撮れる”というざん新なシリーズです。「レゴスタジオ」の核となるのは『レゴ&スティーブン・スピルバーグ　ムービーメーカーセット』(以下，ムービーメーカーセット）です。



▲セットに付属する専用のPCカメラ。内蔵のマイクで音声の取り込みも可能。接続はUSB。ピントの調整は手動

そう，その名のとおり，あの映画監督のスティーブン・スピルバーグ氏が協力した製品です（氏の収益はStarbright基金，Shoah基金に寄付されるとのこと)。

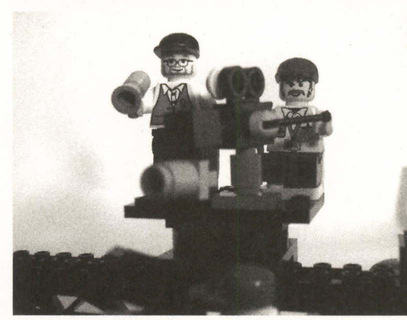
「ムービーメーカーセット」は，映画のセットや役者を模したレゴブロック，PCカメラ，CD-ROM，マニュアルで構成されています。レゴブロックはミニフィグや動物の他に，乗り物や建物，道路，恐竜の足，PCカメラを固定する土台などが用意されていて盛りだくさんです。



▲レゴブロックは全部で433パーツ。カメラマン，音声さん，監督，俳優，女優，消防士，扇風機係，などのミニフィグの他に，ネコ，恐竜などの動物も

レゴスタジオ
レゴ&スティーブン・スピルバーグ
ムービーメーカーセット
希望小売価格29,000円
実売価格23,000円

動作環境
- Pentium 233MHz以上
- OS Windows98
- RAM 32M以上
- CD-ROMドライブ
- USBポート



▲なんとなくスピルバーグ似のミニフィグ（左）がラブリー！

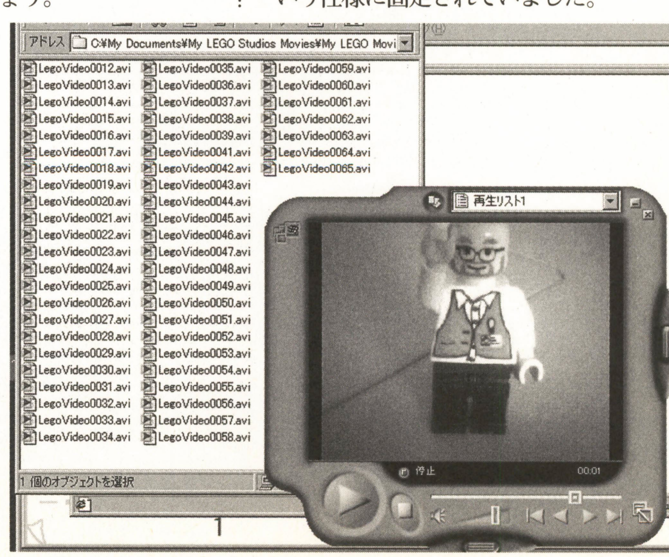
撮　影

まず素材作りのため，PCカメラをパソコンに接続して撮影を行います。PCカメラはライブ撮影の他に，コマ撮りも可能です。たとえば，「ミニフィグが歩くシーン」を撮りたい場合，少しずつミニフィグを動かしながら1枚1枚撮影していきます。コマ撮りは15枚で1秒に相当しますので，3秒なら45枚分シャッターを切らないといけません。かなり根気が必要ですので，がんばって撮影しましょう。

なお，生成されたムービーファイルは15fps，352×288ピクセル，24bitカラー，コーデックはindeo5という仕様に固定されていました。



▲録画の開始はPCカメラのシャッターボタンか，パソコン上の操作で行う

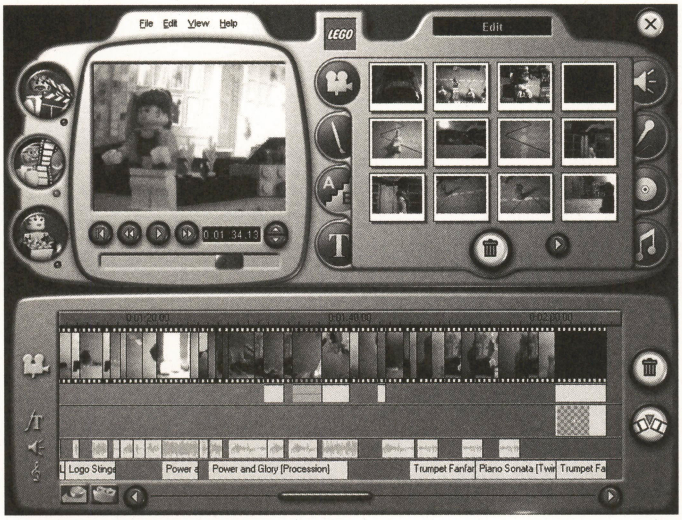


▲撮影したムービーファイルは“My Document”フォルダに蓄えられる

編　集

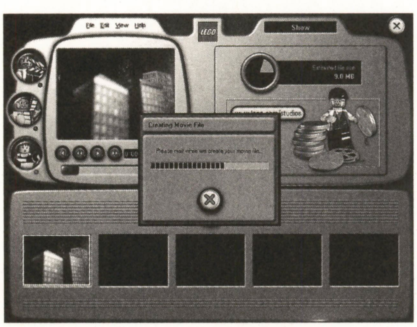
撮影したカットをつなぎ合わせます。編集の方法は簡単で「タイムライン」というウインドウにムービーとサウンドの素材をドラッグで並べていくだけです。編集は1フレーム（1/30秒）タイミングで調節することができます。映像はワイプ処理や，テロップ（字幕）を入れたりすることができます。サウンドはボリューム調整によってフェードインやフェードアウトができます。

この操作性は『Adobe Premiere（プレミア）』にも匹敵する見事な出来映えなのですが，ムービーとサウンドのトラック数が極端に少ないので業務用レベルの凝った編集はできません。



▲編集中のようす。画面下のタイムライン・ウインドウには，びっしりとムービーとサウンドが並んでいる

映画鑑賞


▲レンダリング中。生成されるムービーファイルはMPEG-1形式

完成したムービーはリアルタイムに再生することができます。出来上がった映画はすぐさま編集モードで修正することも可能です。撮影から編集，再生まで一つのソフトウエアに統合されていて，非常に使い勝手がいいです。

出来上がった作品はレンダリング機能で，一つのムービーファイルとして生成してしまうことも可能です。これで作品をホームページで公開したり，CD-Rに焼いてみたりして楽しむことができます。

映画作りはおもしろい!!

技術の進歩によって，こんなに簡単で安価に映画が作れてしまうなんて，非常にすばらしいことだと思います。もう少しPCカメラの画質が上がってくれたら，という気もしないでもないんですが，まあこれだけ充実していて2万円代という価格は大変魅力的です。

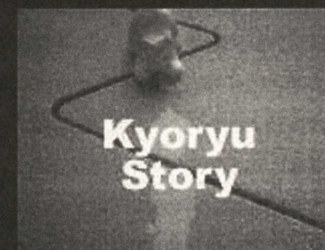
この恵まれた環境から，未来のスピルバーグが現れるかもしれません。「ムービーメーカーセット」はツクモロボコンマガジン館や全国のおもちゃ屋さんで販売中です。興味のある人は店頭で実物を見て検討してみてはいかがでしょう。



＊本記事は海外版『(#1349) The LEGO & Steven Spielberg MovieMaker Set』をもとに執筆しました。　（松原拓也）

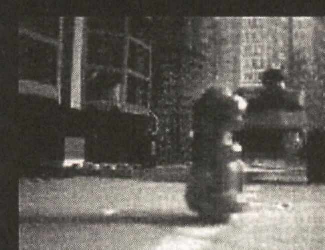
◆ 完　成 ◆

僕もムービーを作ってみました。全65カット，作品時間2分の超大作です。

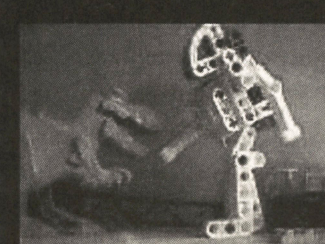


▲オープニング。テロップも思いのまま


▲主人公です。コマ撮りで動きます


▲車にひかれそうな恐竜の子供


▲イロイロあって恐竜が巨大化


▲そしてASIMOとのバトルに発展！ 続きはベーマガのWEBサイトで見てね

松原拓也の自作一代記

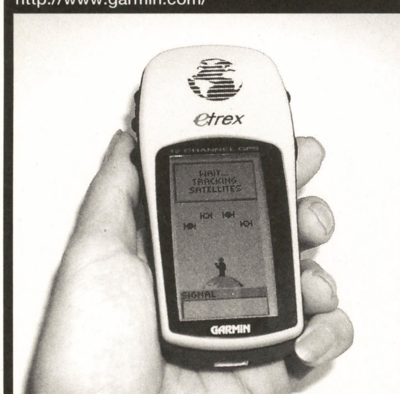
連載5回目 GPSでロボットを自動操縦

「レゴ・ナビゲーション・システム」の製作



GPS（Global Positioning System）はカーナビなどにも応用されている，人工衛星を使った位置測定技術です。今回はそのGPSを搭載して，自動的に目的地へ向かっていくロボットを作りたいと思います。果たしてうまくいくでしょうか。　（松原拓也）

GARMIN「eTrex」（英語版）
実売価格　16,800円（税別）
寸　法　50×110×30mm
重　量　150g（電池込み）
受信部　D-GPS対応，12衛星並列受信
電池寿命　最高22時間（セーブモード時）
GARMIN　WEBサイト
http://www.garmin.com/



GPSを扱っているショップ
■I.D.A（アイ・デイ・ーエー）
http://www.ida-japan.co.jp/
■SPA（システムプロデューサアソシエイツ）
http://www.spa-japan.co.jp/

GPSの購入

GPSのメーカーで代表的なのはソニーとGARMIN（ガーミン）です。価格と性能，開発のしやすさから考えて，GARMINのGPSパーソナル・ナビゲーター「eTrex」を購入しました（写真1）。

「eTrex」は地図機能こそありませんが，トラック表示が可能な12衛星対応のGPSです。防滴機能付きで，大きさは手のひらサイズ。そしてRS-232Cのシリアルポートを通常装備しているのが魅力的です。購入価格は16,800円（税別）。一昔前では考えられない安さです。

「eTrex」を使ってみる

「eTrex」はGPS衛星との距離を計算して，「緯度」「経度」「高さ」「方位」を算出します。「eTrex」の液晶画面には現在位置が「N35° 41′ 33.8″　E139° 42′ 14.7″」という感じに表示されます（写真2）。これはDMSというフォーマットで，「北緯35度41分33.8秒，東経139度42分14.7秒」という意味です。

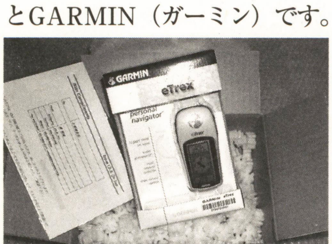
地球儀を見るとよくわかりますが，「北緯」は赤道から北半球に向かう角度で，「東経」は原点から東向きに数えた度数です（第1図）。

通信コネクタの自作

「eTrex」に接続するコネクタを自作します（写真3）。GARMIN製のコネクタの形状は特殊で，純正の通信ケーブルは非常に高価です。そこで「SPA」というお店で扱っている「ePlug」という自作用のパーツがオススメです。僕の場合は形状が似ていた携帯電話（So101）の充電器のコネクタを分解して流用しました（写真4，5，6）。

以下のようにDsub9pinのコネクタ（秋月電子通商にて購入。購入価格50円）とリード線で接続するだけで通信ケーブルは完成です。

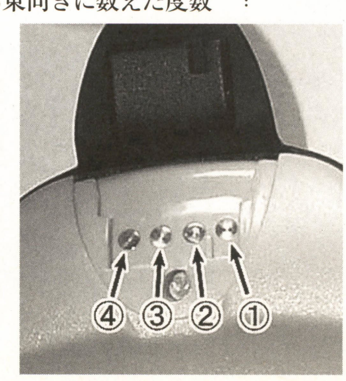
eTrex（専用コネクタ4pin）	←→	PC（Dsub9pin）
①Vcc	ー	接続せず
②Rx	←	③Tx
③Tx	→	②Rx
④Gnd	ー	⑤Gnd



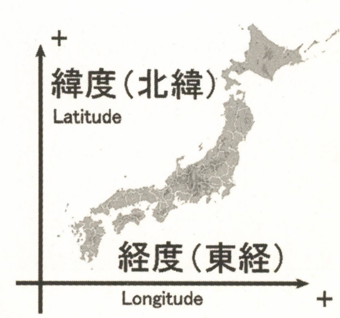
▲《写真1》「eTrex」は通信販売で「I.D.A」というお店から購入



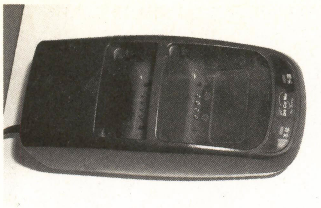
▲《写真2》「eTrex」の動作中のようす。緯度・経度や方位以外にも，移動の軌跡を表示させることができて便利



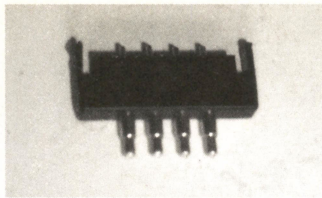
▲《写真3》「eTrex」の通信コネクタ。通信仕様はRS-232Cですごく便利



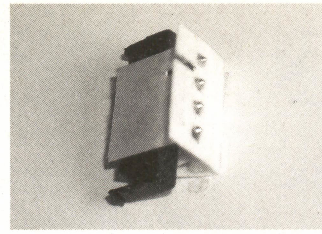
▲《第1図》緯度・経度の概念図。北半球に位置する日本では一般的に「北緯」「東経」を使って度数を表現する



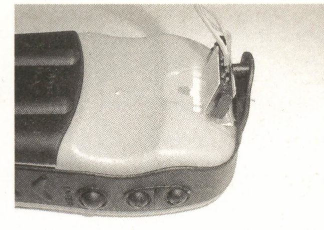
▲《写真4》不要品のSo101（携帯電話）用充電器



▲《写真5》充電器から引き抜いたコネクタ。先端がバネで収縮するスプリングプローブが使われている



▲《写真6》プラ版を加工して接着剤で貼り合わせ，専用コネクタを自作してみた



▲《写真7》「eTrex」にコネクタを取りつけたようす。ぴったりとハマった

「eTrex」とパソコンを通信ケーブルで接続します（写真7）。公開されているフリーウエアを使えば，トラックやウェイポイント（目印）をパソコン上で編集することができるようになります（画面1）。

通信プロトコル

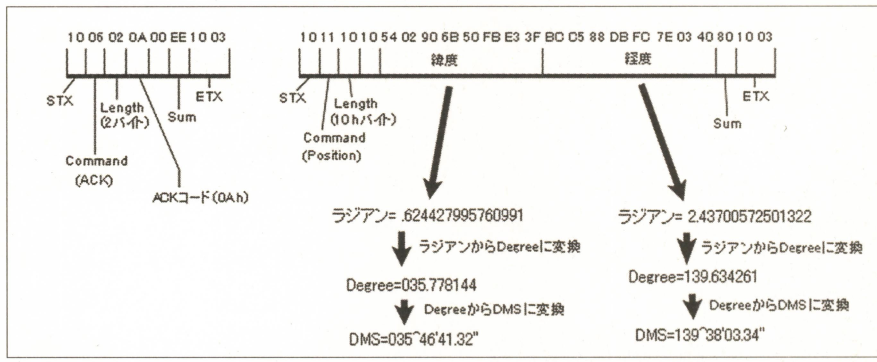
まず「eTrex」の「MENU→SETUP→INTERFACE」を選択して，通信条件を「GARMIN」というプロトコルに設定します。パソコン側は9600bps／パリティなし／データビット8bit／ストップビット1bitに設定します。そして現在位置を取得したい場合，以下の8バイトを送信します。

```
+0 : 10h 'STX
+1 : 0Ah 'Send Position
+2 : 02h 'Length
+3 : 02h
+4 : 00h
+5 : F2h 'SUM
+6 : 10h 'ETX1
+7 : 03h 'ETX2
```

通信フレームは常にSTX（10h）ではじまり，ETX（10h，03h）で終わります。サムはSTXを除く2バイト目からの合計をビット反転した値で，ETXの直前に付加されています。この8バイトを送ると，「eTrex」から31バイトが返ってきます。最初の8バイトが正常終了（ACK）を意味していて，残り23バイトが緯度・経度の情報です（第2図）。

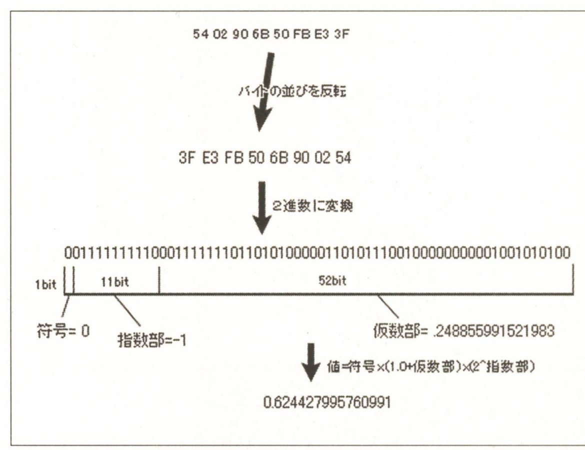


▲《画面1》フリーウエアの「Garmap」でトラックを取り込むこともできる（画面は僕の数週間分の行動範囲）



▲《第2図》パソコンが受け取った31バイトの例。緯度・経度は最終的にDMSに変換される

緯度・経度はIEEE（アイ・トリプル・イー）で決められている64bit浮動小数点（double型）で，最上位ビットから以下のように割り振られています。



▲《第3図》IEEE仕様64bit浮動小数点フォーマットの例

- **符号（全1bit）**
  値の符号です。0でプラス，1でマイナスです。
- **指数部（全11bit）**
  負の値を表せるように，バイアスとして1023足されています。
- **可数部（全52bit）**
  少数で表現された2進数です。最上位ビットが1/2を表し，つぎが1/4，1/8……と，2のn乗の逆数になっています。

これらの値を

符号×（1.0+指数部）×（2^可数部）

という式で計算すると値を求めることができます（第3図）。この変換のために引数に8バイトのバイナリデータを与えると，double型のラジアンを返す“fncStr2Dbl”という関数を作りました。

ラジアンは以下の式で度数に変換できます。円周率の有効数字が多いほど精度がよくなります。

度数＝180×ラジアン÷円周率

数学の授業のようで辟易（へきえき）しますが，まだ計算は続きます。算出した「xx.xxxxxx度」はDegreeと呼ぶフォーマットで，これを「dd度mm分ss.xx秒」というDMSのフォーマットに直します。そのため，DegreeからDMSに変換する“fncDeg2Dms”という関数を作りました。

mm分とss秒は時間と同じく60進数になっています。

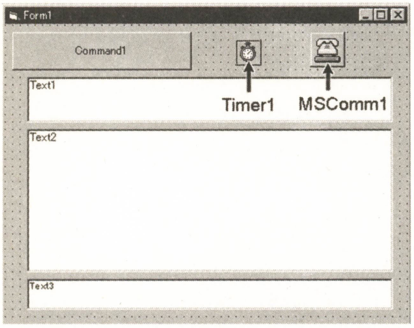
サンプル

VisualBasicを使って簡単なサンプル・プログラムを作ってみました（**画面2**）。ボタンを押すと緯度と経度がテキストボックスに表示されます（**画面3**）。

ロボットの組み立て

LEGO MINDSTORMSを使ってロボットを組み立てます。4kgのノート・パソコンが載っても平気なように，ロボットはできるだけ丈夫で力強いものにしました（**写真8，9，10**）。ビームの補強には「斜め組み」（2001年6月号の「レゴ道場」を参照）を多用しています。

IRタワー（RIS1.0/1.5に付属するシリアル仕様）にデータを送ることによって，パソコンからRCXに接続されたモータを制御します。IRタワーとの通信設定は2400bps／パリティODD／データビット8／ストップビット1です。IRタワーの直接制御については2001年2月号の「レゴ道場」を参照してください。

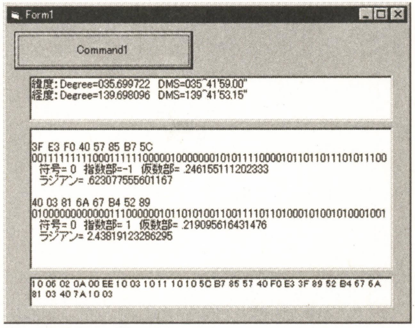


▲《画面2》通信プログラムのフォーム（VB4.0）

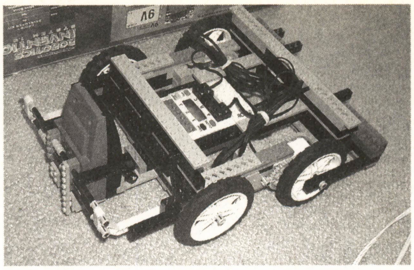
自動走行のアルゴリズム

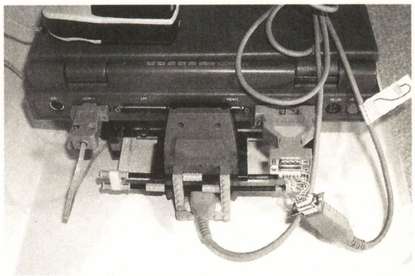
ロボットを前進させる前と後で距離を比較して，目的地までの距離が縮んだらそのまま前進，距離が縮まなかったらロボットの向きを転換するようにしました。現在地から目的地までの距離は三平方の定理を使いました。

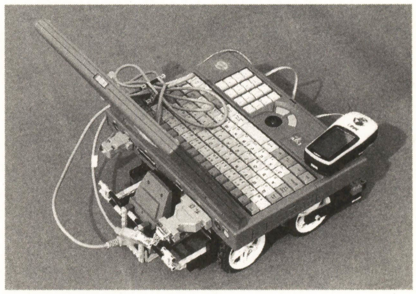
この方法では最適な向きを求めていないので，少々行き当たりばったりに動きます。改造では角度計算の処理を追加するといいいかもしれません。



▲《画面3》実行結果

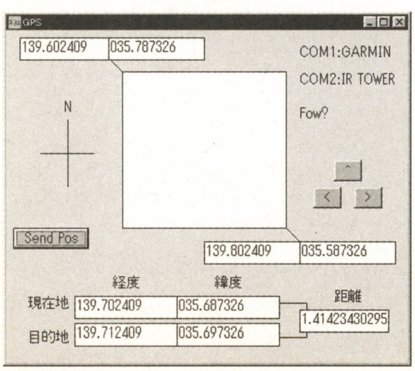

▲《写真8》LEGO MINDSTORMSのRISで作ったロボット


▲《写真9》シリアルポートを二つ装備したノート・パソコンなので，IRタワーと「eTrex」を接続することができる


▲《写真10》完成した「レゴ・ナビゲーション・システム」

走行テスト

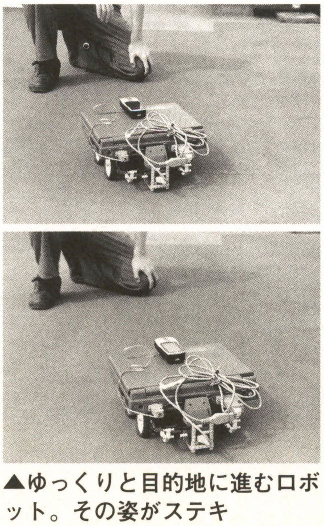
目標地点の緯度と経度をパソコンに入力して，プログラムを実行します（**画面4**）。非常に動きが遅いですが，きちんと向きを修正しています。「レゴ・ナビゲーション・システム」の完成です。

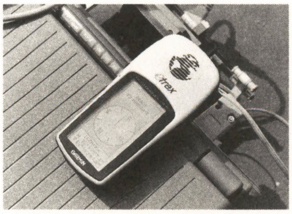


▲《画面4》「レゴ・ナビゲーション・システム」用プログラムの実行画面

こうしてロボットにパソコンを載せる方法はいろいろと応用がききそうです。たとえば，VisionCommandのカメラを載せて画像処理をさせたり，迷路を解くような複雑な思考を行うロボットなどもおもしろそうです。

それと，さすがにノート・パソコンは重いので，PICやAVRでスタンドアロン化させるといいかもしれません。


▲ゆっくりと目的地に進むロボット。その姿がステキ


▲走行中のGPSの値。現在の緯度・経度が表示される


▲電波新聞社の屋上でテスト走行の準備中！

■GPSについて

GPS衛星はアメリカの国防総省が運用しています。GPSは軍事上の理由から「SA（Selective Availability）」という処置で，位置の精度が落とされていました。そして2000年5月，ついにそのSAが解除されました。これによって50mほどあった測定誤差が一気に5m程度になりました。この画期的に向上した性能は「eTrex」でも確かめることができます。

GPSは，少なくとも三つ以上のGPS衛星から電波を受信しなくてはいけません。衛星からの電波は直進性が強いため，室内はもちろん，建物が盃立している場所では電波は届きません。広くて見とおしのいい場所でないと使えないのがGPS最大の泣き所なのです。

《リスト1》緯度・経度表示のサンプル・プログラム・リスト（Visual Basic用）

```
'-----------GARMIN GPS受信機から現在位置を取得するサンプル
Dim sRx As String    '受信バッファ

Private Sub Command1_Click()
   Const BUFFMAX = 8   '送信バッファサイズ
   Dim bTx(BUFFMAX) As Byte '送信バッファ
   Dim iSum As Integer

   Text1.Text = ""
   bTx(0) = &H10 'STX
   bTx(1) = &HA
   bTx(2) = &H2
   bTx(3) = &H2
   bTx(4) = &H0 '
   iSum = &H100 - (&HFF And (bTx(1) + bTx(2) + bTx(3) + bTx(4)))
   bTx(5) = iSum    'SUM
   bTx(6) = &H10 'ETX1
   bTx(7) = &H3  'ETX2
   MSComm1.Output = bTx

   Text1.Text = ""
   Text2.Text = ""
   Text3.Text = ""
End Sub


Private Sub Form_Load()
   MSComm1.CommPort = 1    ' COM番号
   MSComm1.Settings = "9600,n,8,1"   'ボーレート
   MSComm1.Handshaking = 0    'ハンドシェーク
   MSComm1.InputMode = comInputModeBinary
   MSComm1.PortOpen = True    ' ポートを開きます。

   Timer1.Interval = 1
End Sub

Private Sub Timer1_Timer()
   Dim sRx1 As String
   Static iSeq As Integer
   Dim dLatitude As Double
   Dim dLongitude As Double

   If MSComm1.InBufferCount = 0 Then
      Exit Sub
   End If

   MSComm1.InputLen = 1
   sRx1 = MSComm1.Input

   Select Case iSeq
   Case 0: '---------------------STX受信
      If AscB(sRx1) = &H10 Then
         sRx = sRx1
         iSeq = iSeq + 1
      End If

   Case 1: '---------------------ETX受信1
      sRx = sRx + sRx1
      If AscB(sRx1) = &H10 Then
         iSeq = iSeq + 1
      End If

   Case 2: '---------------------ETX受信2
      sRx = sRx + sRx1
      If AscB(sRx1) <> &H3 Then
         iSeq = iSeq - 1
      Else
         For i = 1 To LenB(sRx)
            Text3.Text = Text3.Text + Right$("00" + Hex$(AscB(MidB$(sRx,
i, 1))), 2) + " "
         Next

         Select Case (AscB(MidB$(sRx, 2, 1)))
         Case &H6   'ACK
         Case &H11 'POSITION
            dLatitude = fncRad2Deg(fncStr2Dbl(MidB$(sRx, 5, 8))) '緯度
            dLongitude = fncRad2Deg(fncStr2Dbl(MidB$(sRx, 13, 8))) '経度

            Text1.Text = Text1.Text + "緯度："
            Text1.Text = Text1.Text + fncDeg2Dms(dLatitude)
            Text1.Text = Text1.Text + Chr$(13) + Chr$(10)

            Text1.Text = Text1.Text + "経度："
            Text1.Text = Text1.Text + fncDeg2Dms(dLongitude)
         End Select
         iSeq = 0
      End If
   End Select

End Sub
```

```
'---------------Degreeから DMS(度、分、秒)に変換する関数
Function fncDeg2Dms(dDeg As Double) As String

   Dim sTmp As String
   Dim iD As Integer
   Dim iMin As Integer
   Dim dSec As Double

   sTmp = "Degree=" + Format$(dDeg, "000.000000  ")
   iD = Int(dDeg)

   dDeg = dDeg - iD
   dDeg = dDeg * 60#
   iMin = Int(dDeg)

   dDeg = dDeg - iMin
   dDeg = dDeg * 60#
   dSec = dDeg

   sTmp = sTmp + "DMS=" + Format$(iD, "000^") + Format$(iMin, "00'") +
Format$(dSec, "00.00''")

   fncDeg2Dms = sTmp
End Function

'---------------ラジアンから度数に変換する関数
Function fncRad2Deg(dRad As Double) As Double
   fncRad2Deg = 180 * dRad / 3.141592
End Function

'-------------8バイトバイナリからdoubleに変換する関数
Function fncStr2Dbl(sDbl As String) As Double
   Const PACKBIT = 8    '1バイトのビット数
   Const DBLBYTE = 8    'double型のバイト数

   Dim iSisu As Integer '指数部
   Dim dKasu As Variant '仮数部
   Dim dFugo As Variant '符号
   Dim dTmp As Variant
   Dim bdat(PACKBIT * DBLBYTE) As Variant '2進数テンポラリ
   Dim i, j, k As Integer

   Text2.Text = Text2.Text + Chr$(13) + Chr$(10)

   k = 0
   For i = (DBLBYTE - 1) To 0 Step -1
      ibin = AscB(MidB(sDbl, i + 1, 1))
      Text2.Text = Text2.Text + Right$("00" + Hex$(ibin), 2) + " "

      For j = (PACKBIT - 1) To 0 Step -1
         If (2 ^ j) And ibin Then
            bdat(k) = 1
         Else
            bdat(k) = 0
         End If
         k = k + 1
      Next
   Next

   Text2.Text = Text2.Text + Chr$(13) + Chr$(10)
   For i = 0 To (PACKBIT * DBLBYTE) - 1
      Text2.Text = Text2.Text + Hex$(bdat(i))
   Next
   Text2.Text = Text2.Text + Chr$(13) + Chr$(10)

   dFugo = 1 - (2 * bdat(0))    '符号(1ビット)
   Text2.Text = Text2.Text + "  符号=" + Str$(iSisu)

   '  指数部(11ビット)
   iSisu = 0
   For i = 11 To 1 Step -1
      If bdat(i) Then iSisu = iSisu + (2 ^ (11 - i))
   Next
   iSisu = iSisu - 1023 'バイアスを引く
   Text2.Text = Text2.Text + "  指数部=" + Str$(iSisu)

   '  仮数部(52ビット)
   dKasu = 0
   For i = 12 To (12+52-1)
      If bdat(i) Then dKasu = dKasu + (2 ^ (11 - i))
   Next
   Text2.Text = Text2.Text + "  仮数部=" + Str$(dKasu)
   Text2.Text = Text2.Text + Chr$(13) + Chr$(10)

   dTmp = dFugo * (1 + dKasu) * (2 ^ iSisu)

   Text2.Text = Text2.Text + "  ラジアン=" + Str$(dTmp)
   Text2.Text = Text2.Text + Chr$(13) + Chr$(10)
   fncStr2Dbl = dTmp
End Function
```

テクノ・トイズ・ラボラトリ
Techno Toys Laboratory

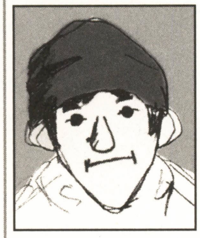
デジタルとアナログの
境界線上で作る電子オモチャ

今月の製作

トータル・ライフ・アドバイザ（後編）

テクノ・オモチャ研究所

TEXT&製作●クワクボリョウタ



デバイス・アーチスト。1998年頃から「ビット」をキーワードにメディアアート作品や電子オモチャなどを製作・発表。ニューヨークのギャラリーPS1で開催される「Buzz Club」という展覧会に出品することに。LSIゲームっぽい作品を製作中デス。詳しくはまた今度。www.vector-scan.com

先月号の回路は完成したかな？ 今月はさとし系電光掲示板「トータル・ライフ・アドバイザ」製作の後編。プログラムを作って，ボディの加工をし，完成させよう。

先月のおさらい トータル・ライフ・アドバイザとは？

トータル・ライフ・アドバイザは，遊ぶ人が「YES」か「NO」で答えられるような質問を念じながら本体を振ると，質問に対するアドバイス・メッセージが表示される。アドバイスといっても実際はあらかじめ何種類か用意したテキストをランダムに選んで表示するだけのインチキなものなんだ。だから時にはトンチンカンな返答があったりするんだけど，またそこが面白いところ。

今回は乱数の出しかた，メッセージ表示方法を中心にプログラムを解説して，最後に筐体の加工について触れるぞ。

ランダム値を得るには

コンピュータでランダムな値，つまり乱数を得るにはどうしたら良いだろうか。コンピュータは正確な演算は得意だけど，逆に言えば正確なことしかできないから，デタラメな値を作りだすのは不得意分野だといえるかもしれない。そこで今回の製作では，起動時に人が本体を振るタイミングを利用して乱数を得ることにした。

前回も説明したように，トータル・ライフ・アドバイザは普段は節電してスリープ・モードになっており，人が本体を振ると内蔵されたチルト・スイッチの状態が変化して起動するようになっている。振るとチルト・スイッチはON状態とOFF状態を交互に繰り返す。

ある程度振ったら表示が始まるようにするために，最初のスイッチの状態の変化でPICをウェイク・アップし，一定時間内に10回振られたことをカウントしたら，メッセージ表示を開始することにしよう。そして，PICのウェイク・アップからメッセージ表示開始までの間，乱数用カウンタを高速にインクリメント（1を加算すること）しておく。こうすれば，メッセージ表示開始のときには乱数用カウンタには毎回違う値が入ってくる。人が本体を振る速さはどんなに正確にやったつもりでもマイコンからみれば十分に誤差があるからだ。メッセージ表示開始までのフローチャートは**第1図**のようになる。

乱数用カウンタには0～255が入ってくるので，下位4ビットを使い，0～15の値を使う。メッセージは16個用意しておき，この値が指し示すものを表示することにする。これでランダムなメッセージを選ぶことができる。

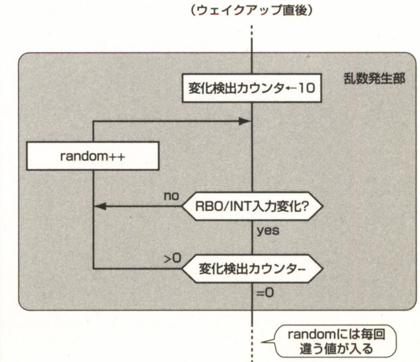
なお，このルーチンでは一定時間スイッチの状態が変化しなかった場合，再びスリープに入るように考慮している。こうしないと，10回振られるまではいつまでも待機してしまい，電池を浪費してしまうからだ。

フォントを作る

トータル・ライフ・アドバイザの表示画面は5×5ドット。ここにカタカナをスクロール表示する。文字は右端から現れ，左に流れていくものとする。メッセージは16種類用意し，先に説明した乱数発生ルーチンから受け取ったメッセージ番号に対応するものを表示する。

テキスト表示と一言でいってしまったけど，それには当然フォントのビットマップ・データが必要になる。残念ながらPICにはそんなもの内蔵されているわけがアリマセン。したがって，50音すべてを自分で作る必要がある。**第2図**が僕がデザインした5×5カタカナのフォント一覧だ。「ア」～「ン」の46文字に「゛」（濁点）と「゜」（半濁点）それに「！」と「？」とスペースを加えた計51文字を使うこととし，「ャュョッ」などの小文字カタカナは大文字で代用することにした。

乱数発生部のフローチャート



《第1図》本体が振られて，チルト・スイッチの状態が10回変化するまで待つ。その間，randomを加算し続けることで乱数を得る

これらのフォントは第3図のルールに従って数値に直し，テーブルとしてプログラム中に埋め込まれる。点を描画するところを「1」それ以外を「0」とした2進数を16進数に直してリストに記述する。横一列ではなく，縦一列に区切って数値化しているのは，横スクロール表示のためにはこのほうが簡単に処理できるからだ。

テーブルの利用

PICの場合，こうしたテーブルを使用してデータを取りだすためには，「retlw」命令を使う。「retlw」命令とはWレジスタに0～255の任意の値を代入してリターンする命令だ。この命令があるアドレスをコールしてあげると，リターン時にはWレジスタに必要な値が代入されてくる。

フォントデータは一文字あたり5バイトで，51文字で255バイトだから，この「retlw」命令を255個ズラッと並べて記述しておく。そして必要なフォント・データの格納されているアドレスを計算し，そこをコールすればフォント・データが取りだせる。

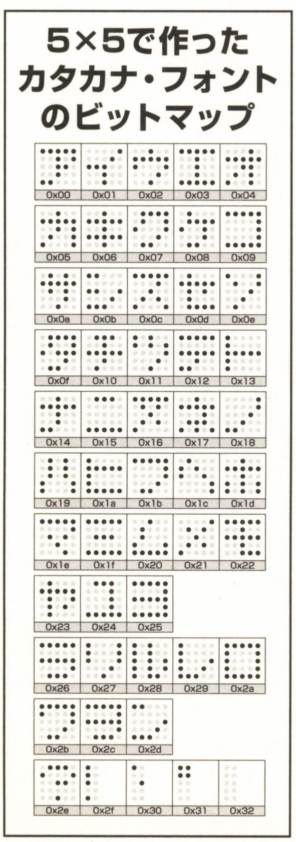
アドレスを計算してコールするには第4図のような方法を用いる。PICの「call」命令は固定したアドレスしか指定できない。そこでプログラム・カウンタであるPCLレジスタに任意のアドレスを代入する方法を用いる。PCLレジスタは実行アドレスの下位8ビットを示す特殊レジスタだ。PCLレジスタに値を書き込むと丁度「goto」命令を実行したようなことが起きる。

ただし，PCLは8ビットのレジスタだから，指定できるアドレスは下位8ビット分だけに限られる。一方，PIC16F84のプログラム・メモリ空間は0x000～0x3ff（16進表記）で，すべてのエリアをカバーするには最低10ビットは必要だ。では上位2ビットはどうすればいいのかというと，これはPCLに書き込む前にPCLATHレジスタに値を書き込んでおけばよい。PCLATHレジスタとPCLレジスタはペアになっていて，これを組み合わせることで最大13ビットのアドレスを指定できる。

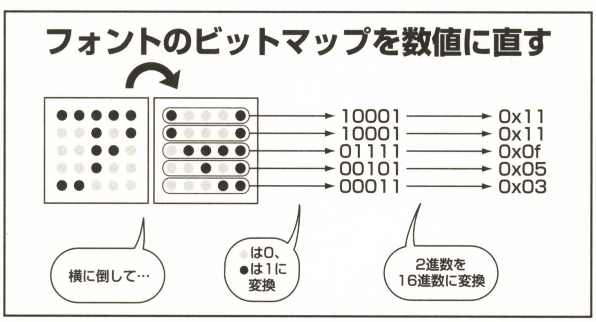
ではもう一度第4図に注目してもらって，計算コールの仕かたを順を追って説明しよう。font_tableと名前の付けられたアドレス0x300～0x3feにフォント・データが並んでいて，その中のひとつ，0x34dにあるデータを取りだしたいとする。まず，PCLATHにテーブルの置いてあるアドレスの上位2ビットである3を代入。つぎに，Wレジスタに取りだしたいデータの置いてあるアドレスの下位バイトを代入。例では0x4dだ。そしてfont_vectorをコールする。font_vectorルーチンはWの内容をPCLに代入。するとつぎのサイクルではPCLATHとPCLの組み合わせである0x34dにジャンプする。0x34dには「retlw 0x19」という命令があるので，Wレジスタに0x19を代入してリターンする。リターン先はどこかというと，もちろんcall命令をしたつぎのアドレスだ。これで無事データをゲットしたぞ。ワカッタ？

メッセージを作る

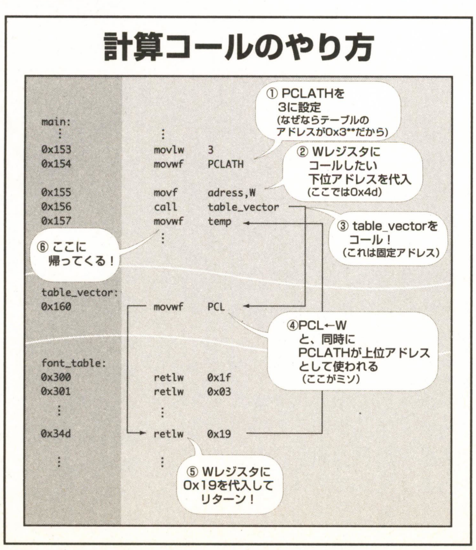
アドバイス・メッセージは16種類用意する。たとえば「マッタク　ソノトオリ」とか「ヨク　ワカラナイナ」とかね。ひとつのメッセージは16文字以内で記述する。そして一文字ずつ，第2図の文字コードに置き換え，これをもとにテーブルを作る。変換の例として第5図を見てほしい。16文字より少ない場合は残りの文字をスペースで埋めておく。したがって，メッセージのテーブルは16文字×16メッセージで256バイトになる。これも先に説明した計算コールのやり方で一文字ずつ読みだして使うことになる。プログラム



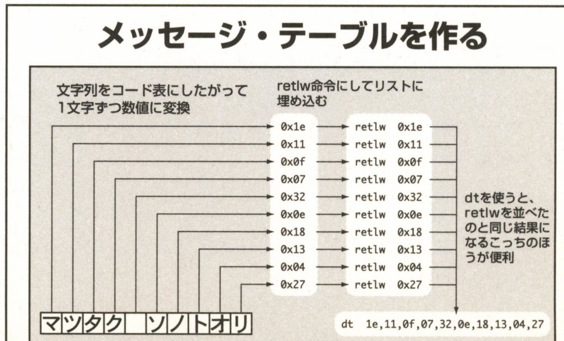
《第2図》それぞれに文字コードを付けて表記してある。ただし，0x2f～0x31は文字幅を2ドットとする



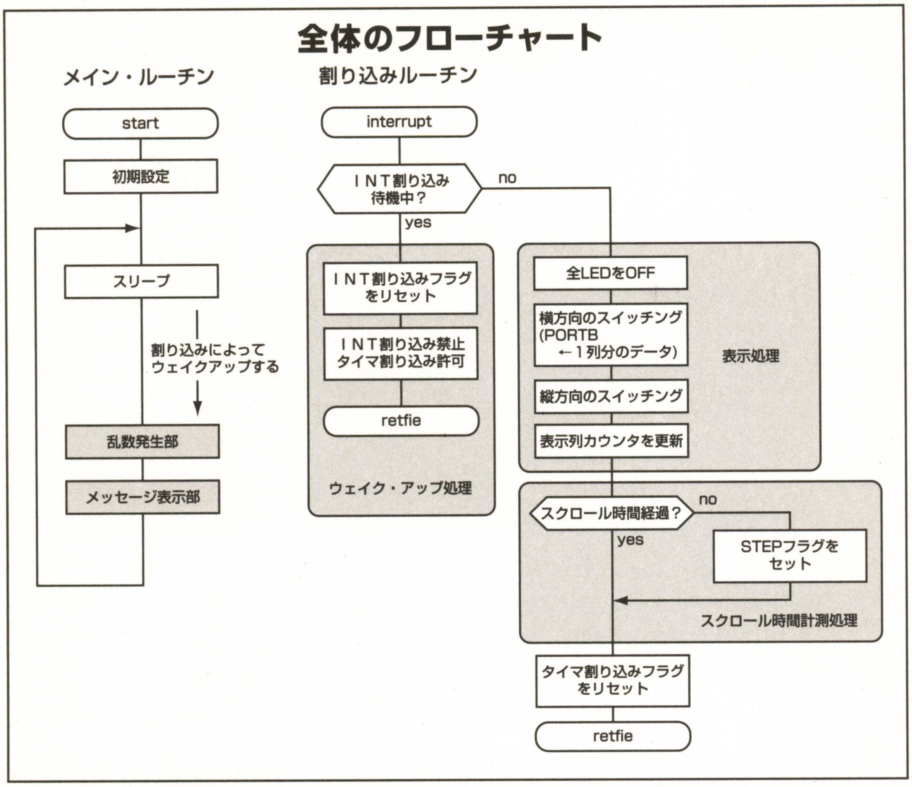
《第3図》横スクロールに都合のいいようにスライスして数値にする



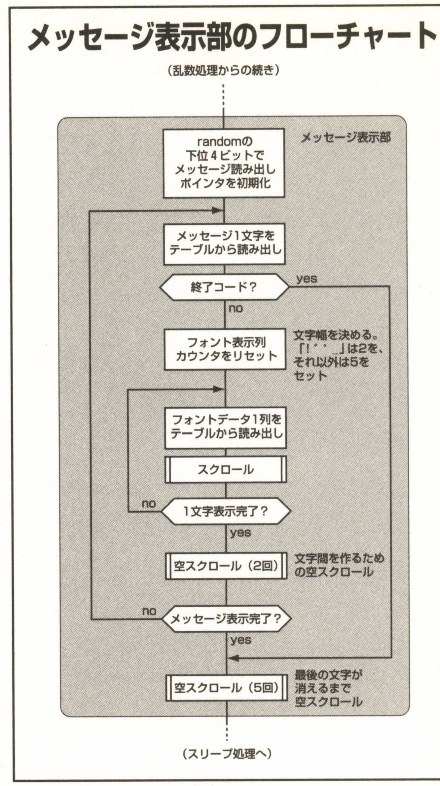
《第4図》PICではアドレスの変化するcallやgotoはこのようにして行う



《第5図》1文字ずつコード表にしたがって数値に変換後，リストの形に直す



《第6図》「乱数発生部」「メッセージ表示部」については第1図，第7，8図に詳しくしるしてある



《第7図》メッセージのテーブルとフォントのテーブル，この二つからデータを読みだしていく様子

中ではテーブルの名前をmessage_tableとした。

このように，メッセージ用とフォント用の2種類のテーブルを使い分けていることに注目してほしい。

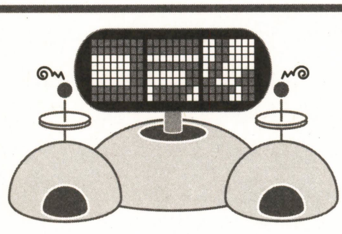
ちなみに16個のメッセージがどんなものかはあえて教えません。これはやっぱり作ってのお楽しみ！ でしょう。・

メッセージ表示の全体像

第6図が全体のフローチャート，**第7図・第8図**がメッセージ表示部の詳しいフローチャート，そして**リスト1**がトータル・ライフ・アドバイザのソース・リストだ。MPASMでアセンブル後，HSモード，パワーアップタイマ使用，ウォッチドッグタイマ不使用にしてPICに書き込む。

プログラムの全体的な流れを説明しよう。

画面表示は前回説明したダイナミック・ドライブによって実現されている。これは割り込み処理によって行われる。disp_bufと，それに続く計5バイトのレジスタ



コンパクトな録音装置で声や音を集めて，パソコンで簡単に編集できるのがこのコンピュータサウンドモーファだ。
この装置，最大4分間の音を記録・再生できる。もちろんそれだけではなくって，パソコンと接続することでいろいろな遊びかたが用意されている。

たとえば自分の声をロボットやねずみ（？）の声に変化させたり，エコーをかけたり，さらには用意されたいろんな効果音をつけたりできる。

それから音の波形を見ながら切り貼りして簡単に編集することもできる。この機能は面白いところを取りだしたり，いらない部分をカットしたり，といったときに使える。ハサミボタンを押すと，長い音声データを自動的に区切ってくれ，細かく区切られた波形はクリックすると再生されるので，音を素早く確かめながら編集することが可能だ。このプレヴュー機能はなかなかイイ感じ。

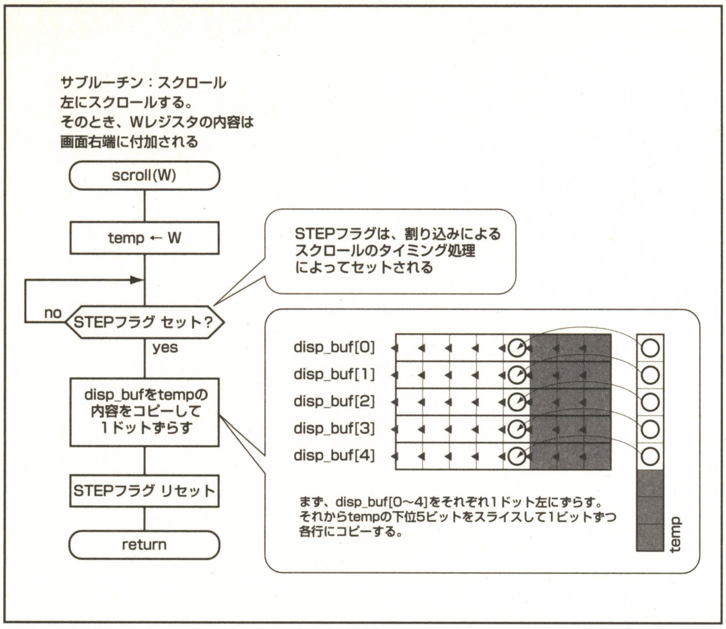
音に反応して動くアニメーションも作れる。Windowsメディアプレーヤでおなじみのエフェクト映像も。顔モードでは福笑いみたいなやり方で顔のパーツを配置すると，再生した声にあわせて口が動く。残念なのは顔の部品の組み合わせが限られていることだ。どうせなら自分の顔をキャプチャして合成できたら面白いと思った。

それからこうして作った音声や，アニメーション付きの音声をそのままメールに添付して送信できる。送信されるのは実行形式の添付ファイルだから，再生に特別なソフトはいらない。天才的な効果音付き一発ギャグを開発して，友達に送るというのも一興かも。

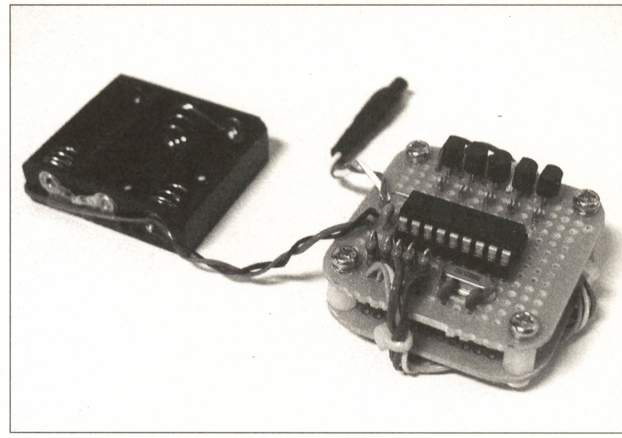


Intel Play
コンピュータ サウンド モーファ

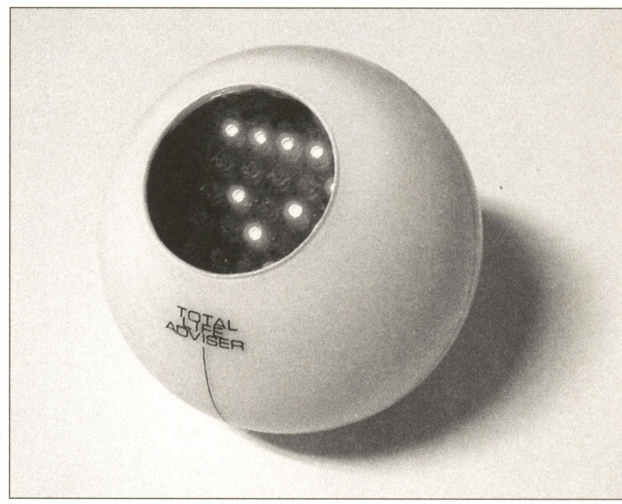
■価格：6,900円
■問い合わせ先
インテル・ホットラインサービス
☎ 0120-868686
■ホームページ
http://www.intel.co.jp/jp/home/products/intelplay



《第8図》フォント・テーブルから読みだしたビットマップ・データを画面表示用メモリにコピーしながらスクロールさせるようす



《写真1》2枚の基板をスペーサを挟んでネジ止めをする。ショートしないように気をつけよう



《写真2》これが完成したトータル・ライフ・アドバイザ。日々迷いが生じたらコイツを振るのだ！

が画面表示用のメモリで，ここにビットマップ·データを書き込んでおくと，自動的に（メインルーチンからは意識することなく）LEDが点灯する。

一方，割り込みルーチンの中ではスクロール速度を得るために一定の間隔で時間計測もやっている。ここでは約0.15秒経過ごとにSTEPフラグを1にセットしている。

スクロールのサブルーチン（**第8図**）ではこのSTEPフラグが1になるまで待機し，1になったら画面更新を実行する。スクロールはdisp_bufの内容を1ドットずつ左にずらす処理をし，新たに表示すべきフォントのデータを右端にコピーすればよい。これをメッセージの長さ分だけ繰り返すと1文表示完了となる。

表示データの取りだしかたは以下のとおりだ。

まず，起動時に得られた乱数の下位4ビットがメッセージ番号になる。メッセージ番号に応じてmessage_tableから文字データを読みだし，その文字データをもとに今度はfont_tableからフォント·データを読みだす。このプロセスは，現時点でフォントの何列目を表示したかをカウントし，またメッセージの何文字目を表示したかをカウントする2重のループになっている。つまり1文字あたり5回のフォント·データの読みだし・画面更新を行い，1メッセージ表示のためにはそれを16文字分繰り返すことになる。

このループから抜けだしたときにはメッセージを表示し終わったことを意味し，マシンは再びスリープに入る。

ボディの加工

最後に，外がわの加工について簡単に説明しよう。これはあくまで製作の一例として参考にしていただきたい。

僕の場合は直径6センチの透明プラスチックの球体を東急ハンズで入手したので，これに3センチの丸穴をあけ，裏面から塗装。つぎに，丸穴の後ろ側からスモーク（半透明のグレイ）のアクリル板を丸く切ったものを接着した。直径は3.2センチだ。これがLEDの表示窓になるのだけど，まっ透明より，スモークや半透明赤色の板を使うと見やすくなる。

そして，基板は球体に収まるように角を削り，2枚の基板をスペーサを挟んでネジ止め（**写真1**）。そしてスポンジテープで適当にクッションをつけて球体に収まったときに程よく固定されるようにした。水銀スイッチは割れると危険なので，ビニル・テープで巻き，衝撃をモロに受けないように固定した。

という具合に完成したのが**写真2**なのだが，もちろん皆さんは入手しやすいものを各自工夫して作っていただければいいのではないかと思う。

最後に

今回の製作はハード，ソフトともにかなり入り組んでいて理解できない人もいるかもしれないけど，まぁいいじゃありませんか。それより，メッセージやフォントをオリジナルなものにして遊んでみてよ。最初は分からなくても改造しているうちにプログラムがきっと自分のものになっていくはずだ！　そして面白いプログラムができたら是非教えてね！

《リスト1》完成版トータル・ライフ・アドバイザのリスト（アセンブラ）

```
;=====================================================================
;
;                total   life   adviser
;
;                 coded  by  ｸﾜｸﾎﾞ ﾘｮｳﾀ
;
;=====================================================================
;
        LIST    C=120, p=16F84, r=dec, x=off, n=0
        include <p16f84.inc>

;======== マクロの定義 ================================================
push    macro
        movwf   w_temp                  ; ﾒｲﾝﾙｰﾁﾝと割り込みﾙｰﾁﾝの両方で
        swapf   STATUS,W                ; 使用しているﾚｼﾞｽﾀを退避させる
        bcf     STATUS,RP0              ;
        movwf   status_temp             ; w_temp      <- W
        movf    FSR,W                   ; status_temp <- STATUS
        movwf   fsr_temp                ; fsr_temp    <- FSR
        movf    PCLATH,W                ; pclath_temp <- PCLATH
        movwf   pclath_temp             ;
        clrf    PCLATH                  ;
        endm

pop     macro
        movf    pclath_temp,W           ; pushで退避させたﾚｼﾞｽﾀの復元
        movwf   PCLATH                  ;
        movf    fsr_temp,W              ;
        movwf   FSR                     ; W      <- w_temp
        swapf   status_temp,W           ; STATUS <- status_temp
        movwf   STATUS                  ; FSR    <- fsr_temp
        swapf   w_temp,F                ; PCLATH <- pclath_temp
        swapf   w_temp,W                ;
        endm

;======== シンボルの定義 ==============================================
;-------- レジスタ名の定義 --------
temp                    equu    0x0c    ; ｲﾛｲﾛな用途
w_temp                  equ     0x0d    ; 割り込み時のﾚｼﾞｽﾀ退避用
status_temp             equ     0x0e    ;        //
fsr_temp                equ     0x0f    ;        //
pclath_temp             equ     0x10    ;        //
y_buf                   equ     0x11    ; 縦方向のｽｲｯﾁﾝｸﾞ位置
count0                  equ     0x12    ; 2秒計測ｶｳﾝﾀ
count1                  equ     0x13    ;     //
event                   equ     0x14    ; 2秒経過の報告用
disp_buf                equ     0x15    ; 画像ﾃﾞｰﾀ格納ﾒﾓﾘ (5ﾊﾞｲﾄ)

prev_portb              equ     0x1a    ; PORTBの変化検出用
random                  equ     0x1b    ; 乱数ｶｳﾝﾀ
shake_count             equ     0x1c    ; あと何回振ったら完了？ｶｳﾝﾀ
letter                  equ     0x1d    ; 表示する文字ｺｰﾄﾞ
letter_ptr              equ     0x1e    ; ﾒｯｾｰｼﾞﾃｰﾌﾞﾙの読み出しﾎﾟｲﾝﾀ
remaining_letters       equ     0x1f    ; ﾒｯｾｰｼﾞの残り文字数
line_ptr                equ     0x20    ; ﾌｫﾝﾄﾃｰﾌﾞﾙの読み出しﾎﾟｲﾝﾀ
remaining_lines         equ     0x21    ; ﾌｫﾝﾄの残り表示列数
bit_count               equ     0x22    ; ｽｸﾛｰﾙ処理用ﾙｰﾌﾟｶｳﾝﾀ


;-------- 各変数の初期値 --------
Y_BUF_INIT      equ     4               ; ﾀﾞｲﾅﾐｯｸ･ﾄﾞﾗｲﾌﾞのｶｳﾝﾀ
COUNT0_INIT     equ     200             ; count0は約0.1秒で0になる
COUNT1_INIT     equ     15              ; ｽｸﾛｰﾙは約0.15秒毎に

;-------- eventのビットマップ --------
STEP            equ     1               ; ｽｸﾛｰﾙ時間経過でｾｯﾄ

;========  リセットベクタ =============================================
        org     0x0000                  ; 起動時は0番地から実行を始める
        goto    initialize

;========  割り込みベクタ =============================================
        org     0x0004                  ; 割り込みが起きると
                                        ;           4番地にｼﾞｬﾝﾌﾟする
;========  割り込みルーチン  ==========================================
        push                            ; ｺﾝﾃｷｽﾄの保存

        btfss   INTCON,INTE             ; INT割り込み許可中なら以下を実行
        goto    int_10                  ; それ以外はint_10へｼﾞｬﾝﾌﾟ

        bcf     INTCON,INTE             ; INT割り込みを禁止
        bcf     INTCON,INTF             ; INT割り込みﾌﾗｸﾞをｸﾘｱ
        bsf     INTCON,T0IE             ; ﾀｲﾏ割り込みを許可

        pop                             ; ｺﾝﾃｷｽﾄ復元
        retfie                          ; 割り込み処理終了

int_10:
        clrf    PORTB                   ; 全LEDを消灯
        clrf    PORTA                   ;

        call    wait_10cycles           ; LEDの消灯のための時間稼ぎ
        call    wait_10cycles           ;

        movlw   disp_buf                ; y_bufが示す1列分の表示ﾃﾞｰﾀを
        addwf   y_buf,W                 ; PORTBにｺﾋﾟｰする
        movwf   FSR                     ;          (横方向のｽｲｯﾁﾝｸﾞ)
        movf    INDF,W                  ;
        andlw   b'11111000'             ;
        movwf   PORTB                   ;

        bcf     STATUS,C                ; y_bufが示す列のﾄﾗﾝｼﾞｽﾀをON
        rlf     y_buf,W                 ;          (縦方向のｽｲｯﾁﾝｸﾞ)
        call    $+2                     ;
        goto    int_20                  ;
        addwf   PCL,F                   ;
        bsf     PORTA,1                 ; (y_buf = 0 の時)
        return                          ;
        bsf     PORTA,2                 ; (y_buf = 1 の時)
        return                          ;
        bsf     PORTA,3                 ; (y_buf = 2 の時)
        return                          ;
        bsf     PORTB,1                 ; (y_buf = 3 の時)
        return                          ;
        bsf     PORTB,2                 ; (y_buf = 4 の時)
        return                          ;

int_20:
        decf    y_buf,W                 ; 次の表示列のために
        movf    y_buf,F                 ;        y_bufを++
        btfsc   STATUS,Z                ;        (0～4の範囲を繰り返す)
        movlw   Y_BUF_INIT              ;
        movwf   y_buf                   ;

int_70:
        decfsz  count0,F                ; ｽｸﾛｰﾙ時間を計測するﾙｰﾁﾝ
        goto    int_90                  ;
        movlw   COUNT0_INIT             ;
        movwf   count0                  ;
        decfsz  count1,F                ;
        goto    int_90                  ;
        movlw   COUNT1_INIT             ;
        movwf   count1                  ;

        bsf     event,STEP              ; 時間経過したらeventﾌﾗｸﾞを立てる

int_90:
        bcf     INTCON,T0IF             ; ﾀｲﾏ割り込みﾌﾗｸﾞをｸﾘｱ
        pop                             ; ｺﾝﾃｷｽﾄ復元
        retfie                          ; ﾀｲﾏ割り込み処理終了

;======== ウェイトのためのサブルーチン ================================
wait_10cycles:
        nop
        nop
        nop
        nop
        nop
        nop
        return

;========  初期設定  ==================================================
initialize:
        clrf    PORTA
        clrf    PORTB

        bsf     STATUS,RP0              ;

        movlw   b'00000000'             ;
        movwf   TRISA                   ;

        movlw   b'00000001'             ;
        movwf   TRISB                   ;

        movlw   b'00011000'             ; TMR0の設定
        movwf   OPTION_REG              ;  (1命令ｻｲｸﾙごとに++)

        bcf     STATUS,RP0

        movlw   0x50-(temp+1)           ;zero clear 0x50 ~ 0x0d
        movwf   temp                    ;
        movlw   (temp+1)                ;
        movwf   FSR                     ;
clr_lp: clrf    INDF                    ;
        incf    FSR,F                   ;
        decf    temp,F                  ;
        btfss   STATUS,Z                ;
        goto    clr_lp                  ;

        movlw   Y_BUF_INIT              ; y_bufの初期化
        movwf   y_buf                   ;

        call    reset_counter           ; ｽｸﾛｰﾙ時間計測ｶｳﾝﾀをﾘｾｯﾄ

        bsf     INTCON,T0IE             ; ﾀｲﾏ割り込みを許可
        bsf     INTCON,GIE              ;
        ;
;========  メインルーチン  ============================================
main:
        clrf    PORTB                   ; 全LEDを消灯
        clrf    PORTA                   ;

        bcf     INTCON,T0IE             ; ﾀｲﾏ割り込み禁止
        bcf     INTCON,INTF             ; INT割り込みﾌﾗｸﾞをｸﾘｱ
        bsf     INTCON,INTE             ; INT割り込みを許可

        sleep                           ; ｽﾘｰﾌﾟ
        nop                             ;

        call    get_random              ; 10回振り終えるまで乱数を更新
        btfsc   STATUS,C                ; 時間切れだった場合は
        goto    main                    ;     mainに戻り、再びｽﾘｰﾌﾟ

        call    display_message         ; ﾒｯｾｰｼﾞ表示

        goto    main                    ;

;======== ｽｸﾛｰﾙ時間計測ｶｳﾝﾀをﾘｾｯﾄするｻﾌﾞﾙｰﾁﾝ ===========================
reset_counter:
        movlw   COUNT0_INIT             ; count0をﾘｾｯﾄ
        movwf   count0                  ;
        movlw   COUNT1_INIT             ; count1をﾘｾｯﾄ
        movwf   count1                  ;
        bcf     event,STEP              ; STEPﾌﾗｸﾞをｸﾘｱ
        return

;=====================================================================
;       サブルーチン : get_random
;       RB0/INTピンの入力が10回変化するまで、randomをカウントし続ける
;       ただし、一定時間変化がなかった場合は、Cフラグをセットしリターン
;       メインルーチンではCがセットされていた場合、スリープに移行する
;=====================================================================
get_random:
        call    reset_counter           ; 時間切れを測るためにﾀｲﾏをﾘｾｯﾄ

        movf    PORTB,W                 ; まず現在のPORTBの値を読み込む
        movwf   prev_portb              ;

        movlw   10                      ; 10回振ったら終了とするため、
        movwf   shake_count             ;     ｶｳﾝﾀを10にｾｯﾄ

get_random_10:
        bsf     STATUS,C                ; STEPがｾｯﾄされていたら、
        btfsc   event,STEP              ;     時間切れを意味する
        return                          ; Cをｾｯﾄし、ﾘﾀｰﾝ

        call    wait_10cycles           ;

        incf    random,F                ; 乱数のｲﾝｸﾘﾒﾝﾄ

        movf    prev_portb,W            ; RB0/INTﾋﾟﾝに変化があるか？
        movwf   temp                    ;
        movf    PORTB,W                 ;
        movwf   prev_portb              ;
        xorwf   temp,F                  ;
        btfss   temp,0                  ; あったら以下の処理へｽｷｯﾌﾟ
        goto    get_random_10           ; なければ繰り返し

        call    reset_counter           ; 時間切れﾁｪｯｸのﾀｲﾏを元に戻す
        decfsz  shake_count,F           ; 指定回数振られたらﾘﾀｰﾝ
        goto    get_random_10           ; まだだったら繰り返す

        bcf     STATUS,C                ; C=0で正常終了を示す
        return                          ;

;=====================================================================
display_message:
        call    reset_counter           ; ｽｸﾛｰﾙ時間のﾀｲﾏをﾘｾｯﾄ

        swapf   random,W                ; ﾒｯｾｰｼﾞ読み出しﾎﾟｲﾝﾀを初期化
        andlw   b'11110000'             ; letter_ptr <- random * 16
        movwf   letter_ptr              ;

        movlw   16                      ; 残り最大文字数は16
        movwf   remaining_letters       ;

l_loop:
        movlw   message_table>>8        ; letter <-
        movwf   PCLATH                  ;   ﾒｯｾｰｼﾞﾃｰﾌﾞﾙから1文字読み出し
        movf    letter_ptr,W            ;
        call    table_vector            ;
        movwf   letter                  ;

        movf    letter,W                ; 終了ｺｰﾄﾞ(0x33)?
        sublw   0x33                    ;
        btfsc   STATUS,Z                ;
        goto    end_of_message          ; だったらend_of_messageへｼﾞｬﾝﾌﾟ

        bcf     STATUS,C                ; ﾌｫﾝﾄ読み出しﾎﾟｲﾝﾀを初期化
        rlf     letter,W                ; line_ptr <- letter * 5
        movwf   temp                    ;
        bcf     STATUS,C                ;
        rlf     temp,F                  ;
        movf    letter,W                ;
        addwf   temp,W                  ;
        movwf   line_ptr                ;

        movf    letter,W                ; ﾌｫﾝﾄの未表示列数の初期化
        sublw   0x2e                    ; remaining_lines
        movlw   5                       ;      <- 「!ﾞﾟ_」は2、それ以外は5
        btfss   STATUS,C                ;
        movlw   2                       ;
        movwf   remaining_lines         ;

s_loop:
        movlw   font_table>>8           ; W <- ﾌｫﾝﾄ1列分のﾃﾞｰﾀを読み出し
        movwf   PCLATH                  ;
        movf    line_ptr,W              ;
        call    table_vector            ;

        call    scroll                  ; scroll(W)
        bcf     event,STEP              ;

        incf    line_ptr,F              ; 1文字表示終了なら以下へ
        decfsz  remaining_lines,F       ;
        goto    s_loop                  ; まだならs_loopから繰り返し

        clrw                            ; 文字間のための空ｽｸﾛｰﾙ
        call    scroll                  ;
        call    scroll                  ;

        incf    letter_ptr,F            ; ﾒｯｾｰｼﾞ表示終了なら以下へ
        decfsz  remaining_letters,F     ;
        goto    l_loop                  ; まだならl_loopから繰り返し

end_of_message:
        clrw                            ; 最後の文字を
        call    scroll                  ;     画面から追い出すまで
        call    scroll                  ;     空ｽｸﾛｰﾙ
        call    scroll                  ;
        call    scroll                  ;
        call    scroll                  ;

        return

;=====================================================================
scroll:
        movwf   temp                    ; temp <- W

        btfss   event,STEP              ; ｽｸﾛｰﾙ時間が経過するまで待機
        goto    $-1                     ;

        movlw   disp_buf                ; ﾙｰﾌﾟ処理の前準備
        movwf   FSR                     ;     FSR <- disp_buf
        movlw   5                       ;     bit_count <- 5
        movwf   bit_count               ;

scroll_loop:
        bcf     STATUS,C                ; 1行を1ﾄﾞｯﾄ左へずらす
        rlf     INDF,F                  ;

        rrf     temp,F                  ; tempの内容に応じて
        btfsc   STATUS,C                ;     右端にﾄﾞｯﾄを打つ
        bsf     INDF,3                  ;      (または打たない)

        incf    FSR,F                   ; ｱｸｾｽする画面ﾒﾓﾘを変更

        decfsz  bit_count,F             ; ﾙｰﾌﾟ終了なら以下へ
        goto    scroll_loop             ; まだならscroll_loopから繰り返し

        clrw                            ; STEPﾌﾗｸﾞをｸﾘｱ
        bcf     event,STEP              ;

        return

;=====================================================================
table_vector:
        movwf   PCL                     ; ﾃｰﾌﾞﾙへのｱｸｾｽ用

;=====================================================================
;       ﾒｯｾｰｼﾞの文字ｺｰﾄﾞ (16ﾒｯｾｰｼﾞ各16文字)
;       文字とｺｰﾄﾞの対応は以下の通り
;       | 00 : ｱ | 01 : ｲ | 02 : ｳ | 03 : ｴ | 04 : ｵ |
;       | 05 : ｶ | 06 : ｷ | 07 : ｸ | 08 : ｹ | 09 : ｺ |
;       | 0a : ｻ | 0b : ｼ | 0c : ｽ | 0d : ｾ | 0e : ｿ |
;       | 0f : ﾀ | 10 : ﾁ | 11 : ﾂ | 12 : ﾃ | 13 : ﾄ |
;       | 14 : ﾅ | 15 : ﾆ | 16 : ﾇ | 17 : ﾈ | 18 : ﾉ |
;       | 19 : ﾊ | 1a : ﾋ | 1b : ﾌ | 1c : ﾍ | 1d : ﾎ |
;       | 1e : ﾏ | 1f : ﾐ | 20 : ﾑ | 21 : ﾒ | 22 : ﾓ |
;       | 23 : ﾔ | 24 : ﾕ | 25 : ﾖ |
;       | 26 : ﾗ | 27 : ﾘ | 28 : ﾙ | 29 : ﾚ | 2a : ﾛ |
;       | 2b : ﾜ | 2c : ｦ | 2d : ﾝ |
;       | 2e : ? | 2f : ! | 30 : ﾞ | 31 : ﾟ | 32 : _ |
;---------------------------------------------------------------------
        radix   hex
        org     0x200
message_table:
        dt      1e,11,0f,07,32,0e,18,13,04,27,33,33,33,33,33,33
        dt      25,07,32,2b,05,26,14,01,14,33,33,33,33,33,33,33
        dt      0f,30,2d,08,30,2d,32,12,30,06,1e,0c,2f,33,33,33
        dt      13,30,04,14,2d,0c,05,17,2e,33,33,33,33,33,33,33
        dt      0d,06,15,2d,32,22,12,1e,0d,2d,33,33,33,33,33,33
        dt      1e,10,05,30,01,14,0b,2f,33,33,33,33,33,33,33,33
        dt      10,05,30,01,1e,0c,2f,33,33,33,33,33,33,33,33,33
        dt      09,30,0b,30,25,02,0f,30,2d,32,12,30,0b,25,2e,33
        dt      0e,29,19,32,00,27,05,22,33,33,33,33,33,33,33,33
        dt      1e,0c,30,32,20,27,12,30,0b,25,33,33,33,33,33,33
        dt      0b,2d,10,25,02,15,32,23,26,17,19,30,33,33,33,33
        dt      05,06,24,02,12,06,32,0c,1f,23,05,15,33,33,33,33
        dt      1e,03,20,06,15,32,08,2d,13,02,0b,0f,01,33,33,33
        dt      0f,30,01,0b,30,25,1b,30,33,33,33,33,33,33,33,33
        dt      22,02,01,11,05,01,33,33,33,33,33,33,33,33,33,33
        dt      0e,29,19,32,19,11,1f,1f,0f,30,2f,33,33,33,33,33

;=====================================================================
;       ﾌｫﾝﾄのﾋﾞｯﾄﾏｯﾌﾟ
;---------------------------------------------------------------------
        org     0x300
font_table:
        dt      11,11,0f,05,03, 04,04,1c,02,01, 06,02,13,0a,06  ; ｱｲｳ
        dt      11,11,1f,11,11, 12,0a,06,1f,02, 12,0a,07,02,1e  ; ｴｵｶ
        dt      0a,0a,1f,0a,0a, 04,13,11,09,07, 04,13,12,0e,02  ; ｷｸｹ
        dt      11,11,11,11,1f, 02,07,12,0f,02, 15,15,10,08,04  ; ｺｻｼ
        dt      11,09,05,0b,11, 02,1f,12,16,12, 01,12,08,04,03  ; ｽｾｿ
        dt      04,13,15,0d,07, 04,15,0f,05,04, 03,00,13,08,06  ; ﾀﾁﾂ
        dt      15,15,0d,05,05, 1f,04,04,04,04, 12,12,0f,02,02  ; ﾃﾄﾅ
        dt      10,11,11,11,10, 11,0b,05,0b,01, 12,0a,1b,0e,12  ; ﾆﾇﾈ
        dt      10,10,08,04,03, 18,07,00,07,18, 1f,12,12,12,12  ; ﾉﾊﾋ
        dt      01,11,11,09,07, 06,01,02,04,18, 1a,02,1f,02,1a  ; ﾌﾍﾎ
        dt      01,05,09,15,03, 11,15,15,15,11, 18,16,11,10,18  ; ﾏﾐﾑ
        dt      10,0a,04,0a,01, 05,05,1f,15,15, 02,1f,02,0a,06  ; ﾒﾓﾔ
        dt      10,11,11,1f,10, 15,15,15,15,1f, 05,15,15,15,0d  ; ﾕﾖﾗ
        dt      07,00,10,08,07, 1f,00,1f,10,0c, 1f,10,10,08,04  ; ﾘﾙﾚ
        dt      1f,11,11,11,1f, 03,11,11,09,07, 05,15,15,15,0f  ; ﾛﾜｦ
        dt      11,11,10,08,04, 02,01,15,05,02, 17,00,00,00,00  ; ﾝ?!
        dt      01,04,00,00,00, 03,03,00,00,00, 00,00,00,00,00  ; ﾞﾟ_
;=====================================================================
        end
```

初心者でもかんたんラクラク作れちゃう！！



iモード・ホームページ講座

第5回　CGIで読者参加型のホームページを作ろう

ホームページを「とりあえず作る」ことから始めて「よりカッコよくする」ことを目指してきた本講座も，第5回を数えて，いよいよ頂点を迎えることができました。今回はこれまでのホームページを一転させる，いわばホームページの極意ともいえるCGIについてご紹介します。

これを極めれば，あなたのホームページはよりいっそうよくなり，たくさんの人が訪れてくれるでしょう。

TEXT●白木ミホ



東京産の神奈川育ち。ｉモード関連の執筆が多い。仕事にはパソコンの欠かせぬ私ですが，休日には運動不足解消と趣味を兼ねて居合術を習っております。剣道でなく居合なので刀を振るわけですが，まだまだ初心者の私は刃のついてない模擬刀を稽古に使っています。いつかは実家に眠るあの真剣を振りたいものよと思いつつ，気分は剣士そのもの。しかし実態は原稿のメ切や，給料入金のタイミングといった日常のもろもろに「しのぎを削る毎日」です。

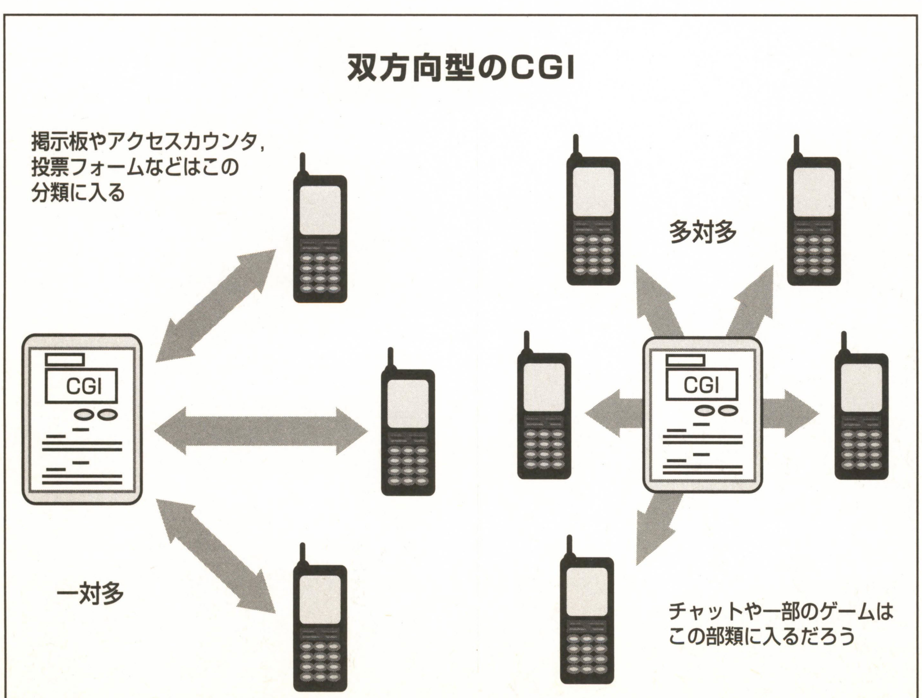
CGIで双方向型のホームページに

ホームページの制作者たちは，次々と作品を発表することではじめは充実した日々を過ごしますが，やがていろいろなことが気になりだします。たとえば，「いったいどのくらいの人が見てくれているのだろう?」「感想は?　面白い?　つまらない?　特にどのへんが?」など。

そうして初めて，今までhtmlテキストを使って作ってきたものは，壁に向かってしゃべっていたのも同然だったと気づき，ガク然とするのです。

そして思うのです。「読者の反応が知りたい」と。その願いをかなえるもの，それが今回のテーマのCGIです。

CGIは作者から読者へと発信されるホームページの世界に，読者から作者へと反応を返すことで，双方向型のホームページを実現してくれる，画期的なプログラムなのです（**第1図**）。



《第1図》作者と読者の間に「一対多」の関係や，「多対多」の関係を作ることができる

CGIの正体

CGI（Common Gateway Interface）とは，perl（パール）言語，またはC言語で書かれるプログラムのことです。htmlテキストの文法であるタグと根本的に違う点は，双方向のやりとりを実現するためにホームページのデータを保存するサーバに，外部からの情報入力（＝書きこみ）を受けつけていることにあります。

ここで代表的なCGIをいくつか紹介しましょう。

●アクセスカウンタ

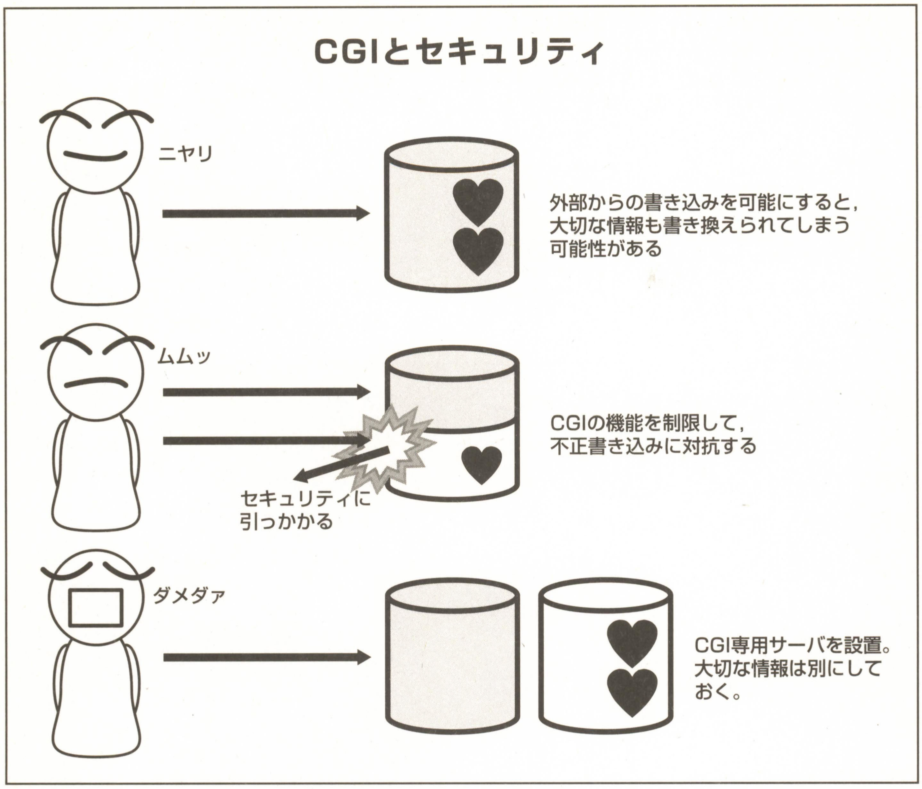
何人の人がそのページにアクセスしてくれたかを表示します。画像タイプとテキストがあります。

●掲示板

好きなときに書きこみができ，他の人の書きこみも同じページに表示される。交流ページ。

●チャット

不特定多数の人とリアルタイムで筆談ができる。



《第2図》セキュリティ面でリスクがともなうCGI。専用サーバを設けたり，アップロード作業を代行してくれるプロバイダもあるが，どれもセキュリティ対策のたまものであり，親切心とは違うのだ

●投稿フォーム
アンケートや投票など形はいろいろ。

●各種ゲーム
おみくじやクイズなど。最近では偶然の妙を楽しむ「文章系」と呼ばれるゲームがはやり。

どれも読者の側から情報提供や参加ができるのが特徴ですが，同時にホームページの中でも格段に人気の高いコーナーになる場合の多い機能といえます。

一般のパソコンでアクセスするホームページでは，今やCGIを使っていないページを見つけることのほうが難しいくらいに，CGIは普及しています。それは，プログラミングをすべて自力で制作する必要のないレンタルCGIや，無料で配布されているCGIがたくさん出まわり，手軽になっていることも背景にあります。

CGI利用の注意点

とはいえCGIの利用には，htmlファイルとは違ったいろいろな制約があります。

あなたのプロバイダでCGIは使用できる?

まず最初にして最大のハードルはズバリこれです。プロバイダによってはCGIの利用そのものが禁止されていたり，使用方法が制限されている場合も多いので，まずプロバイダの利用規約を確認しなければいけません。

制限の理由の一つは，CGIがサーバのセキュリティにリスクを負わせているということです。

CGIは外部からの情報入力を受け付けるプログラムと説明しましたが，それは裏返せば外部からの不正な書き込みに対してもガードが緩くなることを意味します。

実際，htmlファイルでは読み込み許可だけを与えて外部からの書き変えは許可していないので情報が守られています。そのためプロバイダは，CGI専用のサーバを設けて外部からの書きこみを一定の範囲にとどめたり，CGIの機能制限をするなどして情報を守っていることが多いのです（第2図）。

プロバイダによって条件はそれぞれ異なっています。中にはこのCGIの利用規約が新しいプロバイダを選ぶときの目安の一つになるというユーザもいるくらいです。

プログラミングの知識はどのくらい必要か

レンタルCGIのように，一から自作しなくても使えるCGIはありますが，それでもCGIの正体はあくまでプログラム。perl言語の知識がまったくない人には使いこなせないような上級者向けの素材も，もちろんあります。

ですが，少々乱暴ながらも境界線を引くとすれば，本講座でhtmlタグの打ちかたや階層の概念を勉強したあなたなら，簡単なものは十分導入できます。多少難しいものでも親切丁寧な解説のついているものや，ホームページ上にサポート掲示板を設けているところから入門していけば大丈夫。

CGIは重い

CGIの容量は重いので，一般のパソコンで見られるホームページ向けのCGIでは，時にサーバに負担をかけてしまう場合があります。1ページあたりの容量を5kb以下と定めているｉモード用のCGIが，サーバに負担をかけるほど重くなることはないでしょうが，受信する側にとっては，一つのホームページにいくつもCGIが入っていると，情報量の増加＝情報料の増加につながることは考えるべきでしょう。

CGIには専用の設置場所がある

これはCGIを構成するperl言語などを使用できる場所が，サーバの内部で決まっているためです。プロバイダ側の指定してきた場所に間違いなく設置しないとCGIも動きませんし，迷惑もかかります。ファイルや階層の概念がきちんと理解できているかどうかが問われるわけですね。

ｉモード対応のものか？

これが一番重要かもしれません。目的のCGIがｉモードに対応していなければ，せっかく入手しても意味がありません。

CGIをゲットする

さて，プロバイダの規約もＯＫとなったら実際にCGIをゲットしましょう。

プロバイダ自身が推奨する掲示板やチ

ャットといったCGIを利用するのはいちばん間違いがなく，また困ったときにはプロバイダのサポートを受けられるので安心です。これはプロバイダと交渉すればすぐに使用できるでしょう。

CGIのレンタルを専門に扱っているプロバイダを利用するのも一般的です。特に自分の契約しているプロバイダがCGIの使用を禁止している場合は，このレンタルCGI専門プロバイダと契約を結び，掲示板などのスペースを「間借り」して自分のホームページにリンクすればいいのです。これも目的のプロバイダと交渉すればサポートが得られます。

また，無料で配布されている素材をホームページからゲットし，自分なりに書き換えて使う方法もあります。これには多少知識がいるかもしれません。

一般的な手順としては，まず配布を行っているホームページから目的のCGIのファイルをダウンロードして自分のパソコンのハードディスクに保存します。ファイルによっては送信時の容量を減らすために情報が圧縮されている場合もあるので，そのときは解凍を行います。解凍作業には専用のソフトが必要ですので，持っていない場合はこれも無料配布ホームページからダウンロードします。

そうしてファイルを開いて，プログラムの内容を必要に応じて書き換えます。くれぐれも注意書きをよく読んで正しく書き換えてください。

書き換えができたらプロバイダの指定する場所にアップロードします。ただし，アップロードの方法はプロバイダによってそれぞれ指定が異なりますので，これも確認してください。

まとめ

今までの講座と違って今回のCGIには具体的なリストがありません。それは各プロバイダ，あるいは各プログラムによって方法が異なるため，どれか一つを載せても他の場合の参考にはあまりならないからです。CGIの設置はとにもかくにもプロバイダの利用規約を読み，プログラムに添付の説明書きをよく読み，その指示に従うのが唯一の道です。

htmlファイルと違って面倒くさいと思われるかもしれませんが，規約さえ守ればあなたのホームページは格段に面白くなります。それは作るあなたと参加してくれる読者がともに盛り上がって一つのホームページを作り上げていくという，ホームページの醍醐味そのものです。本当に面白いホームページは作者1人の手ではじつはできません。作り手のあなたと，常連と呼ばれる人たちがわいわいおしゃべりし，情報を交換し合うことで「活気」が生まれているということ，それが大事なのです。そういう意味ではホームページも友達同士の付き合いも，根本は同じと言えるでしょう。ただ，それが世界中の人たちとつながる，それがホームページの素晴らしさなのです。

次回は皆さんから寄せられた疑問・質問も交えて，携帯端末ｉモードホームページについてまとめたいと思います。



今回はネットアイドルのランキングや女の子の個人ページへのリンク・情報が満載のページでいろんな女の子の個人サイトをチェックしてみました。

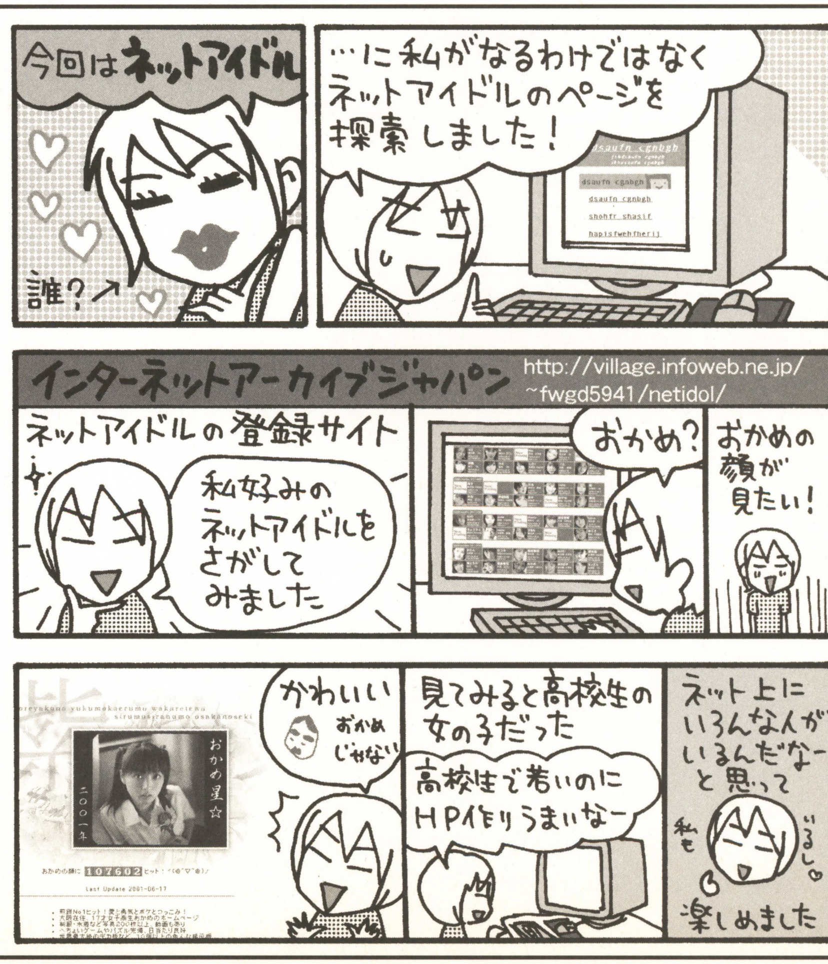
顔写真がずらっと並んでるところから好みの子の写真をクリック，するとプロフィールが出てきて，さらにもっと知りたいと思ったら個人サイトへ行ってみる……という感じで自分の好みの女の子のサイトを探しだす仕組みだ。

そんななか，私は「おかめ」という名前の女の子が気になって個人サイトを見てみたのですが，すごくいいデザインの出来のいいＨＰで，しかも作ってる本人がかなりカワイイときてる超掘りだしサイトでした。

ちなみに男性のネットアイドルのページも見ましたがなんだかホストクラブみたいでした。

しかし，男性のネットアイドルもいたんですね！

（神奈川）



Linux環境でプログラミングを学ぼう！

よいこのLinux修行



第5回 ソフトウエアのビルドとインストール

（※ビルド：ソース・プログラムをコンパイルして，実行プログラムを作成すること）

横井 勝 <masaru@turbolinux.co.jp>

LinuxやUnixでは，多くのソフトウエアがソースの状態で配布されています。利用したいソフトウエアをインストールするとき，ディストリビューションに収録されているバイナリ形式のパッケージを利用する方法もありますが，自分に必要な機能を選んだり，自分でプログラムを修正してビルドしてインストールすることもできます。

せっかくソースが公開されているのですから，そのソースをビルドしてインストールする方法を勉強してみましょう。

プログラミングの方法を解説する書籍などの基本的な部分で，「Hello World」という文字列を出力するプログラム例が紹介されている場合があります。多くのフリーソフトウエアを公開しているGNUプロジェクトでは，Hello Worldを出力するためのプログラムをまじめに公開しています。今回は，このHello Worldを表示するためのプログラムであるGNU Helloを例に，ソフトウエアのビルドとインストールの方法を紹介します。

1. GNU Helloの入手と展開

GNU Helloは，ftp://ftp.gnu.org/gnu/hello/やGNUソフトウエアがミラーされているFTPサイトから入手できます。執筆段階の最新バージョンは1.3で，hello-1.3.tar.gzというファイルがあります。第1図はFTPクライアントのlftpを利用してダウンロードする様子です。各種FTPクライアントを利用してダウンロードしてみてください。

ダウンロードしたhello-1.3.tar.gzを展開(extract)するには，tarというコマンドを利用します。シェルで以下のようなコマンドを実行すると，現在のディレクトリ(current directory)にファイルが展開されます。

```
tar -zxvf hello-1.3.tar.gz
```

すると，hello-1.3という名前のディレクトリにファイルが展開されます。このあとの作業は，このディレクトリで行うので，カレントディレクトリを移動させてください。

```
cd hello-1.3/
```

2. ビルドとインストール

ソフトウエアによってはビルドとインストールの方法が若干異なる場合がありますが，今回は多くのソフトウエアで採用されているconfigureスクリプトを実行したあとmakeしてインストールする方法を紹介します。ビルドとインストールの方法は，そのソースの中のREAD MEなどの説明を行っているファイルなどを参照してください。

2.1. ちょっとその前に…。

このあとでインストールを行うときに，hello-1.3に含まれているファイル"Make

```
$ lftp ftp.gnu.org
lftp ftp.gnu.org:~> cd gnu/hello/
cd ok, cwd=/gnu/hello
lftp ftp.gnu.org:/gnu/hello> ls
-rw-r--r--   1 ftp      ftp         16452 Sep 17  1992 hello-1.0-1.1.diff.gz
-rw-r--r--   1 ftp      ftp         25676 May 22  1993 hello-1.1-1.2.diff.gz
-rw-r--r--   1 ftp      ftp          2919 May 23  1993 hello-1.2-1.3.diff.gz
-rw-r--r--   1 ftp      ftp         87942 May 23  1993 hello-1.3.tar.gz
lftp ftp.gnu.org:/gnu/hello> get hello-1.3.tar.gz
87942 bytes transferred in 2 seconds (43.7K/s)
lftp ftp.gnu.org:/gnu/hello> bye
```

《第1図》GNU Hello を lftp でダウンロードしているようす

file.in"は，そのままの状態ではインストールできない場合があります。"Makefile.in"の21行目あたりを，**リスト1**のパッチを参考に直してください。

2.2 configureを実行する

configureは，そのソフトウエアが必要としているライブラリが利用可能かどうかを調べたり，インストールするディレクトリを設定したり，選択可能な機能を選択することができます。configureを実行するには，そのソースのディレクトリに移動してシェルで以下のように実行します。

```
$ ./configure
```

configureを実行すると，sedによる置き換え機能を利用して，makeコマンドが利用するMakefileをMakefile.inから，利用可能なオプションをC言語の"#define"文で定義したファイルであるconfig.hをconfig.h.inから生成します。

configureは，/bin/shなどのBourne shellで動作するシェルスクリプトになっています。GNU Helloに含まれているconfigureは555行のスクリプトとなっています。configureは，autoconfというコマンドで"configure.in"というファイルをもとに，m4というマクロプロセッサを利用して生成されます。

configureではオプションを指定することができます。しかしhello-1.3に収録されているconfigureスクリプトは古いautoconfが生成したスクリプトなので，今回説明したいヘルプ文に関するオプションが利用できません。現在インストールされているautoconfを使ってconfigureの更新を行ってから，configureのヘルプ文を見てください（**第2図**）。

《リスト1》Makefile.inのパッチ

```
--- hello-1.3/Makefile.in.dist  Tue Jun 12 08:45:44 2001
+++ hello-1.3/Makefile.in       Tue Jun 12 08:46:00 2001
@@ -17,8 +17,7 @@

 #### Start of system configuration section. ####

-VPATH = @srcdir@
-srcdir = $(VPATH)
+srcdir = @srcdir@

 CC = @CC@
```

```
$ autoconf
$ ./configure --help
Usage: configure [options] [host]
Options: [defaults in brackets after descriptions]
Configuration:
  --cache-file=FILE       cache test results in FILE
  --help                  print this message
  --no-create             do not create output files
  --quiet, --silent       do not print `checking...' messages
  --version               print the version of autoconf that created configure
Directory and file names:
  --prefix=PREFIX         install architecture-independent files in PREFIX
                          [/usr/local]
  --exec-prefix=EPREFIX   install architecture-dependent files in EPREFIX
                          [same as prefix]
  --bindir=DIR            user executables in DIR [EPREFIX/bin]
  --sbindir=DIR           system admin executables in DIR [EPREFIX/sbin]
  --libexecdir=DIR        program executables in DIR [EPREFIX/libexec]
  --datadir=DIR           read-only architecture-independent data in DIR
                          [PREFIX/share]
  --sysconfdir=DIR        read-only single-machine data in DIR [PREFIX/etc]
  --sharedstatedir=DIR    modifiable architecture-independent data in DIR
                          [PREFIX/com]
  --localstatedir=DIR     modifiable single-machine data in DIR [PREFIX/var]
  --libdir=DIR            object code libraries in DIR [EPREFIX/lib]

        以下省略
```

《第2図》configureのヘルプ文

このヘルプ文を読むと，"--prefix="のところにディレクトリを指定すると，ユーザーが実行するファイルを，指定されたディレクトリの中の"bin"，または"--bindir="で指定されたディレクトリにインストールして，システム管理者が実行するファイルを，指定されたディレクトリの中の"sbin"，または"--sbindir="で指定されたディレクトリにインストールする，といった設定ができることが読みとれると思います。このconfigureスクリプトがどのようになっているかを知りたい方は，ぜひconfigureスクリプトを読んでみてください。

このあとに，configure を実行してみてください。**第3図**のような結果になると思います。

```
$ ./configure
creating cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc  ) works... yes
checking whether the C compiler (gcc  ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking how to run the C preprocessor... gcc -E
configure: error: can not find install-sh or install.sh in . ./.. ./../..
```

《第3図》configure を実行

最後の行でinstall-sh というスクリプトがないということでエラーになって終了しています。一部のUnix ではinstallというコマンドが収録されていない場合があります。install-sh は，installコマンドのない環境でインストールするために，install コマンドの機能を提供するためのシェルスクリプトなのです。install-sh はTurbolinuxの場合，autoconfのパッケージに含まれている/usr/share/autoconf/install-shにあるので，このファイルを現在のディレクトリに コピーしてください。

$ cp /usr/share/autoconf/install-sh ./

それでは，ファイルをコピーしたあとにもう一度configureを実行してみましょう（**第4図**）。

うまくいったみたいですね。

```
$ ./configure
loading cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc  ) works... yes
checking whether the C compiler (gcc  ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking how to run the C preprocessor... gcc -E
checking for a BSD compatible install... /usr/bin/install -c
checking for ANSI C header files... yes
checking for string.h... yes
checking for fcntl.h... yes
checking for sys/file.h... yes
checking for working alloca.h... yes
checking for alloca... yes
updating cache ./config.cache
creating ./config.status
creating Makefile
```

《第4図》ファイルコピー後に再度，configure を実行

2.3. make を実行する

configureから生成された，Makefileを利用してmakeしましょう。シェルで以下のように実行してください。

make

すると，コンパイラのgccを用いてソースファイルをコンパイルをします。問題なくコンパイルが終わると"hello"という名前のファイルができていると思います。それでは，このhelloが正しく動作するか調べて見ましょう。シェルで以下のように実行してみてください。

$ ./hello
Hello, world!

このように「Hello world!」と表示されれば成功です。このhelloにはいくつかのオプションがあるようです。helloに"-h"というオプションを追加して実行すると**第5図**のような説明文が表示されます。

```
$ ./hello -h
This is GNU Hello, THE greeting printing program.
Usage: ./hello [-htvm] [--help] [--traditional] [--version] [--mail]
  -h, --help                 Print a summary of the options
  -t, --traditional          Use traditional greeting format
  -v, --version              Print the version number
  -m, --mail                 Print your mail
```

《第5図》hello に -h を追加して実行した説明文

2.4. インストール

たんに「Hello world!」なんて表示するプログラムをインストールするというのも変なのですが，練習ということでインストールしてみます。

configureスクリプトでインストール先のディレクトリを設定していない場合，ファイルhelloは"/usr/local/bin/"にインストールされるようになっています。ディレクトリ"/usr/local/bin/"は，ユーザーrootだけが書き込みできるようになっているので，ここにインストールする場合はrootになる必要があります。rootになってインストールするにはシェルで"makeinstall"を実行します。それではrootになってインストールしてみましょう（第6図）。

すると，/usr/local/bin/hello と /usr/local/info/hello.info がインストールされます。

```
$ su
Password: ほげほげほげ (←表示されません)
# make install
./mkinstalldirs /usr/local/bin /usr/local/info
/usr/bin/install -c hello /usr/local/bin/hello
/usr/bin/install -c -m 644 ./hello.info /usr/local/info/hello.info
```

《第6図》rootになってインストールしてみた

3. バイナリパッケージも同じように作れる

Turbolinuxなどのディストリビューションに収録されているバイナリパッケージの多くは，今回紹介したビルドの方法を用いてソースから作っています。Turbolinuxで利用しているRPM (RedHat Package Manager)では，「SPECファイル」と呼ばれる種類のファイルに，ソースの展開，ビルド，インストールの手順をBourne shellのスクリプトで記述するようになっています。

SPECファイルの例の一つとして，wgetというWebサイトのミラーなどを行うツールのSPECファイルを紹介します（第7図）。このSPECファイルは，http://pkgcvs.turbolinux.co.jp/cgi bin/cvsweb.cgi/wget/1.7/wget.specで見られます。

このSPECファイルの"%setup"からあとの部分では，%buildの部分でconfigureスクリプトを実行して，makeして，"%install"の部分でmake installしています。

こんな感じでバイナリパッケージは作られているのです。

```
%setup -q
%patch1 -p1
#%patch2 -p1 -b .skip
%build
./configure --prefix=%{__prefix} --sysconfdir=/etc --enable-ipv6
make

%install

rm -rf $RPM_BUILD_ROOT
make install prefix=$RPM_BUILD_ROOT%{__prefix} sysconfdir=$RPM_BUILD_ROOT/etc
mkdir -p ${RPM_BUILD_ROOT}%{__prefix}/share
mv ${RPM_BUILD_ROOT}%{__prefix}/info ${RPM_BUILD_ROOT}%{__prefix}/share/info
mv ${RPM_BUILD_ROOT}%{__prefix}/man ${RPM_BUILD_ROOT}%{__prefix}/share
gzip $RPM_BUILD_ROOT%{__prefix}/share/info/*
strip $RPM_BUILD_ROOT%{__prefix}/bin/* || :
```

《第7図》wget.specの一部

電子学園50周年記念イベント
『レゴとロボットの未来学』講習会開かれる

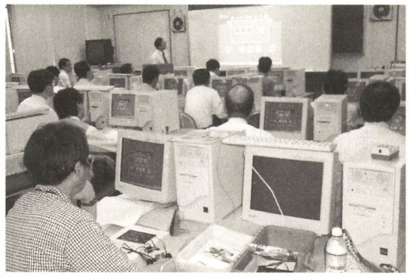
日本電子専門学校創立50周年記念イベントの一環として，関東地区の工業高校の先生方を対象に『レゴとロボットの未来学』という講習会が6月9日に行われました。

この講習会は，LEGO MINDSTORMSによるロボット作りを実際に体験できるというもので，LEGO MINDSTORMS ROBOLABOの販売を手がけるラーニングシステムの協力で行われました。

まだ「LEGO MINDSTORMS」を知らないという先生方が多く，ロボットの組み立てに最初は手間取っていたようですが，数時間後には全員ライントレース（線を追従する）のプログラミングをマスターできたようです。参加した先生方のLEGO MINDSTORMSへの反応も上々で，これを機にLEGOブロックが学校の授業で登場するかもしれません。

なお，日本電子専門学校では，来年4月に「おもちゃ工学科」を新設。この夏の体験入学では，「LEGO MINDSTORMSによるロボット作り」を実際に体験してもらおうという企画を進めているということです。

日本電子専門学校
http://www.jec.ac.jp/
ラーニングシステム
http://www.mdstorm.com/



▲数時間の授業でライントレーサー・ロボットの組み立てからプログラミングまでマスター



▲「おもちゃの未来学」というテーマで講演するエレクトロニクス系・系長の阪井茂さん



◀プログラムしたロボットを実行。緊張の一瞬。先生方も真剣そのもの

日本電子専門学校
一日&半日体験入学

●場所：
日本電子専門学校9号館メディアセンター
●一日体験学習（申込制）
●開催日程：
平成13年
7月20(祝)，26(木)，27(金)，28(土)
8月1(水)，2(木)，23(木)，24(金)，25(土)，
27(月)，29(水)
9月9(日)，23(日)
10月14(日)
●スケジュール：
9：45〜受付開始
15：30〜体験実習終了後，入学相談受付開始
●申し込み方法：
一日体験入学は予約制で，電話またはホームページで受付中です。
TEL 0120-00-9691
WEBサイト
http://www.jec.ac.jp/taiken/onedaysanka.html

●半日体験学習（申し込み不要）
●開催日程：
平成13年
8月22(水)，26(日)，28(火)
9月16(日)
10月21(日)
11月4(日)
12月16(日)
平成14年
1月20(日)
2月3(日)，19(火)，26(火)
3月5(火)，10(日)，19(火)
●スケジュール：
13：00〜
15：30〜 入学相談
●申し込み方法：
申し込みの必要はありません。当日，直接会場に

電子学園50周年記念イベント
レゴ・ロボット競技大会「LEGOおもちゃコンテスト」

●主催：日本電子専門学校
●開催日：10月6日（土曜）午後2時
●会場：新宿NSビルB1　イベントホール
●競技部門：デザイン部門（フューチャー賞，ドリーム賞，LEGO特別賞）＾ムービング部門（車輪タイプ，二足歩行タイプ，多足歩行タイプ，その他）
日本電子専門学校
http://www.jec.ac.jp/

茂木でASIMOとLEGOに出会おう「ASIMO WORLD」

栃木県にある「ツインリンクもてぎ」で，二足歩行ロボットの「ASIMO」をテーマにしたイベントが行われます。

会場では「ASIMO」の歩行デモンストレーションの他に，LEGO MINDSTORMSを使った工作教室も行われます。その教室では，なんと本誌でお馴染みの松原拓也さんや桐林星河くんがインストラクターとして参加する予定です。果たしてどうなることやら。お楽しみに！

●主　催：本田技研工業
開催日：2001年 8月4日(土)〜8月14日(火)
※ただし 8月7日(火)は休み
●会　場：ファンファンラボ(ツインリンクもてぎ)
栃木県芳賀郡茂木町桧山120-1
本田技研工業
http://www.honda.co.jp/
ツインリンクもてぎ
http://www.twinring.co.jp/
ファンファンラボ
http://www.honda.co.jp/fanfunlab/

ツクモカップ「第1回ミニゲーム」

LEGO-MUG（LEGO MINDSTORMS Users Group JAPAN）が「ツクモロボコンマガジン館」の協力を得て，LEGOロボットを使った競技会「第1回ミニゲーム」を行います。見物も歓迎ですのでぜひご参加ください！

●開催日時：7月22日(日)
●開催場所：ツクモロボコンマガジン館3F
●競技内容：ライントレース競技
※詳細は決まり次第，以下のWEBサイトで発表します。

LEGO-MUG
（LEGO MINDSTORMS Users Group JAPAN）
http://www.legomug.gr.jp/

ツクモロボコンマガジン館
http://www.tsukumo.co.jp/shops/robo.html

「第5回 LEGO MINDSTORMS フェスティバル」

LEGO-MUG（LEGO MINDSTORMS Users Group JAPAN）主催の「第5回MINDSTORMSフェスティバル」が東京ビッグサイトで行われます。コンテストでは，多くの力作が大集合します。ぜひご覧ください。

●主催： LEGO-MUG
（LEGO MINDSTORMS Users Group Japan）
●開催日：8月5日（日）
●会場：東京国際展示場「東京ビッグサイト」
東京都江東区有明3-21-1
＊「LEGO MINDSTORMSフェスティバル」は「東京トイフェスティバル」内で開催されます。
LEGO-MUG
http://www.legomug.gr.jp/
東京トイフェスティバル
http://member.nifty.ne.jp/toyfes/
東京ビッグサイト
http://www.bigsight.or.jp/

中学生・高校生を対象にした
ホームページコンテスト開催！

ICF（財団法人　国際コミュニケーション基金）では，「自分発，新コミュニケーション」をテーマに『第4回ICFホームページコンテスト』を開催します。対象は中学生・高校生，まさにベーマガ読者の力の発揮どころです。個人はもちろん，団体での応募も可能です。友達，クラスの仲間，学校など，ふるって応募してください。

詳細は下記のとおり。

●応募期間：2001年7月16日（月）
　～2001年9月30日（日）

●応募資格：中学生・高校生

●応募方法：ホームページ上でURLを登録
（http://www.icf.or.jp/contest/）

●審査：独自性，コミュニケーションへの意思，双方向性，国際性，質・量，見やすさ・扱いやすさなどを基準に，1次審査・2次審査により総合的に評価。

●審査発表：2001年10月19日（金）
　ホームページ上で発表

●問い合わせ先：

e-mail　contest@icf.or.jp
fax　03-5202-8509

ICF
（財団法人　国際コミュニケーション基金）
http://www.icf.or.jp/

平成12年度「第3回」の受賞者

【個人の部】

大　賞	該当者なし
特別賞	吉本龍司
佳　作	北野華子
佳　作	井口智志
佳　作	二木美香

【団体の部】

大　賞	兵庫県立西脇高等学校 生活情報科2年10組 ファッションプロジェクトチーム
特別賞	秋穂中学校　科学部
佳　作	三隅町立井野中学校
佳　作	和歌山県立串本高等学校 商業クラブ
佳　作	亀岳中学校生徒会
佳　作	北海道函館中部高校 パソコン部

レゴの専門店
「0522SHOP（ゼロゴウニイニイショップ）」



今年4月，東京都駒込駅に「0522SHOP（ゼロゴウニイニイショップ）」というお店がオープンしました。「0522SHOP」は，レゴブロックやトミーのゾイドを中心に扱っている珍しいおもちゃ屋さんです。

店長の久保さん自身，相当なレゴ好きで，それが高じて店を自分で作ってしまったそうです。

店内には輸入品の珍しい製品の他に，約30年前のレアな製品が売られていました（国内の最新版も扱っています）。

そして，入り口付近にはレゴブロックのガチャポン（1回200円）が置かれていたり，店長秘蔵のコレクション（非売品）が展示されています。

いたるところに店長さんのこだわりが感じられるお店でした。



「0522SHOP（ゼロゴウニイニイショップ）」
〒114-0024　東京都北区西ヶ原1-54-4
佐々木ビル201
TEL／FAX　03-3917-7112
営業時間　13:00～21:00
定休日　水曜日

※場所はJR駒込駅北口から本郷通りを進み，第一勧銀の交差点を左に曲がってすぐ，佐々木ビルの2階です。

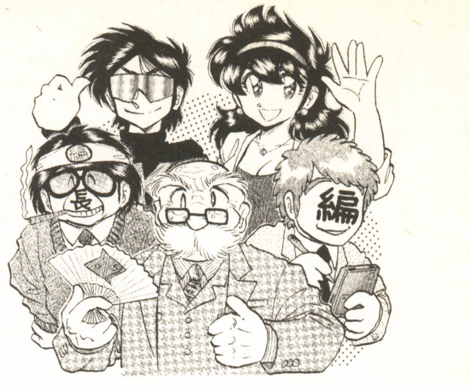


▲「（#5218）Pnumatic Pack」や海外のMINDSTORMS本や，レゴブロックと本が一体になった「LEGO CRAZY ACTION CONTRAPTION」など珍しいものが売られていた



▲30年前のものから，最新のものまで幅広く売られている

あくせすROOM



OOF ダブルオーエフ

編：夏に入って，ますます湿気が気になるところですが，太陽エネルギーを吸収してがんばりましょう！

つぐ美：わーい，夏休み～。

Dr.D：浮かれとらんで，お盆進行に備えるのじゃ。締め切りが早まるのじゃぞ。

影：単なるダジャレではじめた「萌えっ子プロダクション（萌えプロ）」なんだけど，投稿がとても多くて，今月も心なしかイラスト重視なレイアウトとなってしまったよ。

編：と言いますか，1ページはさみしいですね。愛読者カードよろしくお願いします。

♥JAVAやケータイも良いが，ビデオデッキはダメですか？　機能が多くて使いきれん！

（愛知県名古屋市・最馬場可　？歳）

つぐ美：あれは，お父さんお母さんが子供に操作を教えてもらうことで，自然な親子の交流をはかっているのよ。

影：つぐ美ちゅわ～ん，このソフトの操作方法がわかんないんだけど教えてくんない？

Dr.D：どれ，わしが見てやるかの。

♥FM音源のオーケストラヒットが懐かしいこの頃です。

（鹿児島県川内市・純太　？歳）

つぐ美：オケヒって，ジャン！っていう，いろんな楽器の音色が混じったアレ？

編：携帯電話でFM音源の着メロを楽しむなら，やっぱりN503iですよね。配信サイトには，懐かしい曲がごろごろ公開されていますよ。

♥5月15日現在，闘技場の賞品CD-ROMがまだ来ていません。首を長くして待っています。早く発送して（笑）。

（福島市・南斗ライバ　？歳）

影：空腹は最大の調味料。

編：そういえば，電波ビルの地下書庫で，山積みになったベーマガのバックナンバーと格闘してましたね。

影：ああいう紙ばかりの場所にいるとトイレが近くなるのはなぜだろう。

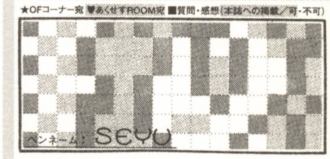
編：空気中から繊維質を摂取してるからじゃないですか？



▲愛知県名古屋市・わいち301
Dr.D：PS2のBASIC記事を目当てに買ってくれたそうじゃ。投稿もたのむぞい。



▲山形県米沢市・転職MAYORSON♡
つぐ美：もしかして合作でくま？　とってもあたたかいくま。また送ってくま～。



▲愛知県刈谷市・せゆ　26歳
つぐ美：1m離れて目を細めて見ると，女の子の横顔に見えるんですって！

現役敗残兵による文章レベルアップ術

全国1億8千万人の作家志望のみなさんコンバンワ。日本人として生まれたからには，猫も杓子も小説を書きたいお年頃。作家になれなかった現役ライターが匿名でガン首並べて，一瀉千里に文章力が増大するテクニックを伝授しようってわけです。（匿名ライターZ）

Q.仕事が忙しくて作品を書くヒマないです。

A.作家で食っていける人はそうそういませんし，会社で得られる情報・知識は重要です。働きながら書く習慣を身につけましょう。

まあ，毎日書くというのが無理ならば，週末などにまとまった時間をとって執筆にあてましょう。一気に書いたほうが良質な作品になる人も多いですしね。

Q.最近，本を乱読・精読しても得られるものがありません。なんとなくスランプです。

A.好きな作家さんの作品を，手で書き写してみましょう。新しい発見があるはずです。手が痛いというなら，ワープロでもいいです。そのかわり，変換できなかった単語はすべて辞書登録しておいてください。使いたい単語がうまく変換できないと，面倒になって別の言葉に代用してしまいがちですが，これはいけません。

素材名人 萌えプロ

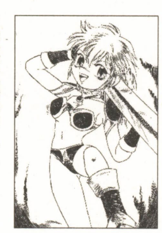
お題を組み合わせて，魅力的なイラストを完成させよう！　今月のテーマはつぎのとおり。

・おねいさま系　・刀剣類　・大きなクツ

あと，3か月ぶんのお題をナコルルに注ぎこんだ大阪府松原市・マヲは反則っーことで。



▲千葉県柏市・鈴木有人
影：これだけ大きいクツだと……エティンとかトロールだったらヤだなあ。



▲愛知県名古屋市・のうぎょうアッパー
編：スタンダードなヒロイック・ファンタジーですね。今月は愛知県民が強いです。



▲和歌山県橋本市・ヅラchu
つぐ美：白足袋にワラ草履。使命のためとはいえ，男になりきれない女のコって感じかな。



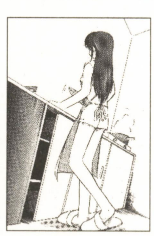
▲神奈川県平塚市・らーま
つぐ美：こういう妹もほしいかなあ。
編：お姉さんっぽさって難しいですね。



▲愛知県丹羽郡・裁刃鋭人
編：素材どうしがぶつからないようにするのがコツですね。これは落ち着いた雰囲気が◎。



▲愛知県名古屋市・わいち301
影：なるほど，これは考えた。
つぐ美：今月の応用カナンバーワン!?



▲埼玉県北足立郡・ぶんちょ
影：個人的にスリッパは，ファンシーな装飾の多いものが楽しくて好きなのだが。

お題と締切日

影：新しいキーワードは，みんなの要望（好み）から選んでるよ。難しい人は，テーマのうち二つだけでもいいので挑戦してみよう。

8月号	9月号	10月号	11月号
・おねいさま系	・お供（ペットや従者）	・妹	・メイドさん
・刀剣類	・サングラス	・せまいところが好き	・楽器が好き
・大きなクツ	・フリルいっぱい（服に限定せず）	・西部開拓時代	・（自由課題）
（今月号で発表）	（締切：2001年6月30日に終了済）	（締切：2001年7月30日）	（締切：2001年8月30日）

あくせすROOM応募のルール

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

ワンポイント
まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

ワンポイント
下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫
〒141-8715　東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

大人気クリエイター集団「ギガ連射」があれこれ語る

GIGA RENSYA つつうらら

•第11回•

ベーマガ読者のみなさんはじめまして。グラフィックデザインのラナスです。桜も梅も季節が過ぎ，夏も近いですね。原稿を書いている今，札幌はまだ微妙に肌寒いので，夜の自転車は半袖だと風邪ひきピンチな今日この頃です。この本が書店に並ぶ頃は「Tシャツでも寒くないぜ！」と親指を立てたイカしたポーズを取れると思います。というわけで，今回は僕がつつうらうらなことを書いていこうかと思います。

『ネタ決め』

僕がゲームの設定や小道具などのネタを決める場合は，いくつかパターンがあります。まぁそんなに大げさなものじゃないですけど。一つは実体験をもとにネタをひねります。実験体ではないです。いろいろ経験したことをネタにする漫画家などわりと多いみたいですが，僕がクターのゲームを作るときにも，体験したことなどをもとにする場合があります。そのまんま使うわけではないですが，多少は体験したことが盛り込まれています。たとえば，気分転換に海岸にシートを敷いて会議をやったことや，昔お祭りで入ったチープなお化け屋敷のことがネタになったりしてます。いざというときネタに困らないように，自分の経験値がたまればたまるほど良いです。

二つめは，時事ネタなんかを盛り込んでみたりします。時事ネタといってもアンダーグラウンドなほうに走っても困りますので，大きなイベントがあったとか，そういう感じのものですね。たとえば僕が押し進めた企画「クターのマキュー」は若鷹軍団優勝記念だったりします。野球はあんまり好きではないのですが，ON対決だったし，食品店行ったらテーマソングが耳についてしまったし，記念のトレーナー買ってたし…。

三つめは好きなもの。一番パッと思い浮かびやすいと思います。サイトで配布してる壁紙や携帯に待ち受け画面は，地元札幌のサッカーチームが好きで，なかなか好調なところから作成しました。好きなものだとなかなか楽しんで作れます。他人の企画はわりとイヤイヤです（笑）。

四つめはボツネタの再利用。次回作のネタに詰まることがよくあります。「早くゲーム出してください」とメールが来ても，「クターの●●」と掲示板で書かれても，早く作れないし，そんなベタベタなネタじゃ…。そんなときは，ボツ企画を掘り起こして，なんとかします。「クターのロケット」なんかはいい例です。企画書用に描いた絵をもとに膨らましていきました。ボツネタは他にもまだまだあるので，何かの機会にゲームとしてお見せしたいと思います。他のゲームについては，メンバーがどうやってネタ出しているのか，実態がよくわかっておりません。夢で見たって人もいるみたいだし。で，僕なりのネタ決めの結論としては，

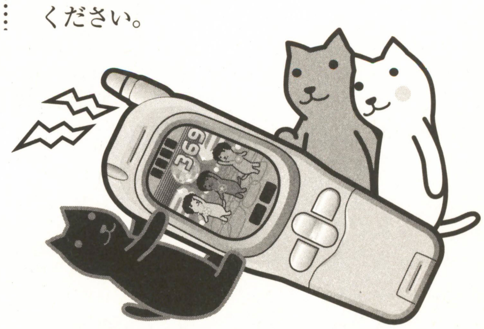
1・経験をしてそれを生かせ！
2・世の中のことにも目を向けろ！
3・好きな物だと楽しんで作れる！
4・資源の再利用も大事！

というわけで，そこんとこよろしく。

『携帯電話について』

とりあえず，実験程度な感じで携帯向けのページなんか作ってみたものの，所詮は「勝手サイト」なので，たいした物は作ってなかったりします。わりと楽しんで画像を作ってますが，それにしてもなんで電話機のメーカーによって画面の仕様がバラバラなんでしょうね？全機種共通にしようとすると94×72ドットなんていう小さく半端なサイズだし，容量も少ないし，白黒画面のやつもいまだにあるし…もう書き始めると切りがないので，このへんでやめときますが，チヤホヤされて喜んでる開発者出て来い！！

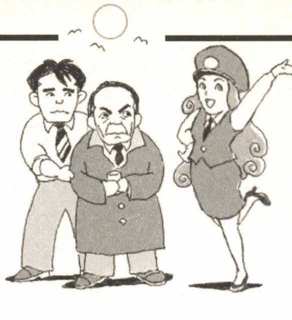
ところで，よくメールや掲示板で書かれるのが，「なんでi-modeだけなの？J-PHONEやAUも作って！」というのがあります。作って作れないことはないです。しかし，メンバーの持っている携帯で，まともにサイトが見られる端末がi-modeしかないんです(笑)。おまけにｉアプリが使える機種も皆無ですので，JAVAゲームなどの公開はもうちょっと先になりそうです。メンバーはメールが使えればよいぐらいにしか思っていないからかもしれません。でもまあ，密かに携帯進出は狙っているので，DOCOMO以外のユーザーさんもちょっとは期待してみてください。



感想・質問・交際の申し込みなど，みなさんからのメールお待ちしています。
otayori@giga-rensya.com

ギガ連射WEBサイト
http://www.giga-rensya.com/

コーエー番デカが贈る
コーエー 窓際デカ最前線



横浜の片隅にひっそりと棲息する，窓際デカ長とその部下達。
その任務とは…謎のゲーム会社コーエーの悪事を暴き出し，事件を未然に防ぐことだった！

株式会社コーエー
http://www.koei.co.jp/

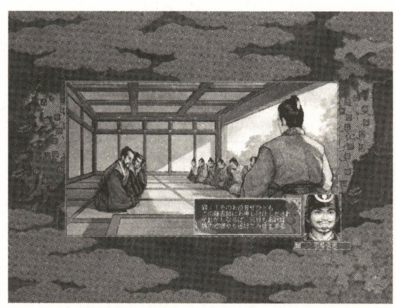
長らくのご愛読ありがとう！ついに感動の最終回！

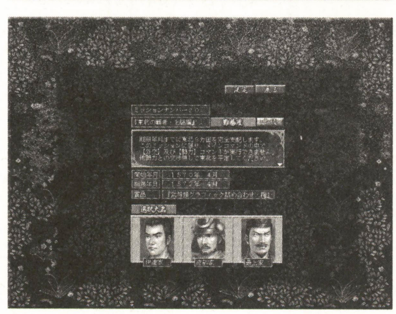
デカ長：この派出所もついにOA化の波に飲み込まれ，閉鎖の憂き目をみることになったワケか…。
刑事3：いや～，OA化に負けてしまう警察官っているんですねえ，本当に。
デカ長：本当にOAのせいなのか？！もしかしたらもの凄い陰謀に巻き込まれているんでは？！
刑事3：それは，気のせいですって。だって，デカ長なんてブラインドタッチ一つマスターできないんですから，現場を退いている管理職としては致命的ですよ。
デカ長：なんだとこのやろ！マシンを使いこなせたって部下を使いこなせるとは限らねえんだ！オレ様の偉大さがわからねえヤツがいけないんだー！えーい，さっさと最後の仕事に行って来やがれコンチクショウ！

報告書

すっかり夏めいてきた。この夏こそは車のボンネットの上で目玉焼きを焼いてみたいものだ。何しろ自分は冒険野郎だからして，大自然のすさまじい威力を目の当たりにするのにも全く臆することはない。だからこそ，その勇気をかわれて，誰一人として帰ってこれないという噂の派出所に栄転されることになったのである。晴れがましい気持ちで一杯だ。

「そこのキミ。先ほどから何をぶつぶつつぶやいているんだい？まあ，わかるよ。熱中してるとついついでちゃうんだよね。「殿！あそこに見えるは桔梗の紋！明智軍でござりまする！」とか「出陣じゃー！槍をもてー！」とか「このえせ茶人が！！」とかね。自分なんて今『信長の野望 嵐世記 パワーアップキット』の制作が佳境なんだもの，どうしたって頭の中身が口の端からボロボロこぼれちゃうのよ。まあその原因の一つとして，今回のパワーアップキットで搭載される「歴史イベントエディタ」があげれるんだけど。シリーズ初で搭載されるこの機能では，イベント条件，登場人物，セリフや背景グラフィックを設定するだけで，自分だけのオリジナルイベントが作れちゃうんだ。まさにオレ好みの謀略謀反イベントなんか簡単に作れちゃうんだよね。しかも友達とメールなどで交換できるんだから，たまりませんなぁ。他にも，期限付きのミッションをクリアすると，その成績によって新家宝や新シナリオが報償として得られる「ミッションチャレンジモード」を搭載したり，シリーズ初の超激ムズ「超級モード」が用意されたりなどなど，そのパワーアップぶりは半端じゃないんだ。どう？キミもチャレンジしたくなっちゃっただろ。パワーアップキットでキミもパワー





アップ！なんてね」

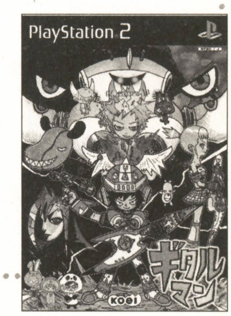
キシャー！これ以上オレ様が強くなったらこの日本はどうなることか！このほとばしる気力と体力と精神力の暴走を止めることは，何人たりともできやしないんだ！！オラオラーそこのけそこのけオレ様が通るぞー！（キキーッ！ドンガラガッシャン！）

デカ長：なにぃー！？車にはねられただとっ！？自爆して終わりなんて，そのまんますぎて悲しいぞ？！コラーーーー！！！

〒223-8503神奈川県横浜市箕輪町1-18-12　（株）コーエー

今月の押収品①

PS2で大好評発売中の青春音楽ゲーム『ギタルマン』



イラストライター326さんみたいに絵が上手ければ，きっと楽しい人生だろうね。羨ましい限りですな。

今月の押収品②

発売中『ギタルマン オリジナル・サウンドトラック』だ。



絵だけじゃなくって音楽もノリノリでイケイケ（死語）なのね。

最終回　婦警さんの㊙情報部屋

はっっぁ～～～い♥♥　みなさんに支えられてぇ～，なんと2年以上！も続いたこのページもぉ，実はいきなり最終回なんで～す♥　イヤ～ン♥　いろんなことを思い出すわネ～♥♥　投稿がちっとも来なくてぇ～，たまぁ～に来るおハガキを眺めて暮らしたりぃ～，一瞬だけタイトルイラストがすげ替わったりぃ～♥　うふふ，楽しい思い出ばっかりでぇ～っす♥　それじゃ最後にイラストをご紹介♥

福島県の葵鉤菜さん，ありがと～♥♥　和風乙女さんたちのバイブル，PSゲーム『遙かなる時空の中で』の源頼久～♥♥♥　Windows版で続編も鋭意制作中なのよネ～♥婦警さんもそろそろ職替えして，巫女さんにでもなろうかしらん♥　でもこれからも，コーエーとぉ，コーエー女子社員とぉ，ゲームの中のイケてる男の子♥は，ずぅ～っとチェックしていきたいと思いまぁ～す♥

長い間ご愛読ありがと～♥　それじゃぁ，またどこかでお会いしましょう～♥♥♥　婦警さんでしたっ♥



▶by　葵鉤菜さん

SEGAのヘンな話題やちょっとお得な情報ならおまかせ!!

ベーセガ

マイコンBASIC SEGA OFFICIAL VARIETY SPACE

奈美　こんにちはぁー，ベーマガ読者のアイドル「らぶりーなっち」ですぅ。

竹崎　な，奈美さん，何か変なもの食べたんか!? それとも便秘か，別人か，うぽろぴこか？

奈美　えへ♪　最近一緒にやっているPSOのお友達に「らぶりーなっち」って呼ばれているんですぅ。今月から私はこういうキャラで行きますので，みんなもよろしくねー☆

竹崎　PSO友達は，実物を直接見たことがないからそんな間違った名前で…。（ドカバキッ！）

奈美　さて，静かになったところで，今月も「ベーセガ」のはじまりでぇーす。

竹崎　いてて…。正直者を殴るなよ～。

奈美　それよりも竹崎さん，なつかしのあの人がゲームになりましたよ!!

竹崎　誰だれ!?　奈美さん教えてよ！

奈美　竹崎さん，ちゃーんと「らぶりーなっち」って呼んでくれないとー。でね，ドリームキャストのCMといえば！

竹崎　CM？　タッキー？

奈美　その名も，U(ゆー)かわ・S(スーパー)しぇんむー!! 略して「USシェンムー」!!

竹崎　長い間，お世話になりました……。

奈美　待ってください～，竹崎さぁーん！

竹崎　よくもそんなダジャレ思いつくなッ！

奈美　「USシェンムー」は，7月5日に3000円で発売になる海外版「シェンムー 一章 横須賀」のことですよー！

竹崎　わかっとるわい！「USシェンムー」は，ボイスや画面内の表記が全て英語に変更されているが，日本語の字幕にも対応してるから英語がダメでも遊べる「シェンムー」さ。

奈美　英語の勉強にもなるというわけですね。

竹崎　待望の「シェンムー2」もいよいよ9月6日に発売されるし。

奈美　おさらいとしては，バッチリですねっ。

竹崎　細かいことだけど「USシェンムー」と「シェンムー 一章 横須賀」のセーブデータは完全互換なのだ。「一章 横須賀」の続きを「US～」で…といった遊びかたもできる。

奈美　あ～それは嬉しいですよね。当然どちらのセーブデータでも「シェンムー2」へ移行できるってことですしー。親切設計♪

竹崎　「シェンムー2」の予習でやっとくしかないって感じか？　そして次の「シェンムー2」には豪華な「バーチャファイター4プレミアムディスク」も初回限定でついてくるという話だよね。期待してしまうぜ～！！！

奈美　おおー，竹崎さんが燃えている～。

竹崎　そう，そして今月はもう1本。奈美さん，いよいよ「アウトトリガー」がでますぜ！

奈美　だから～「らぶりーなっち」ですぅ。

竹崎　ネットワーク多人数対戦が熱いんだよね。

奈美　最大4人まで，インターネット回線で同時対戦できるんですよぉ～。

竹崎　皆で対戦したら面白いぜっ！

奈美　らぶりーなっち…。

竹崎　はぁ？

奈美　らぶりーなっち…。

竹崎　あのな～ッ！

奈美　野沢菜というのは知ってますが，あの菜というのは知りません。

竹崎　あう～っ！とトリガーで，対戦者同士が出会うとどうなるのか知ってるか？

奈美　鳥が出てくるんですよ！

竹崎　その答えは？

奈美　「会うと鳥がー」!!

竹崎　さらば，奈美……。

奈美　だから～「らぶりーなっち」ですぅ。

竹崎　……。

奈美　らぶりー…。（ドカバキッ！）

竹崎　さてさて，この「アウトトリガー」，なんと，マウスに対応しているんだよね。

奈美　マウスで操作するのもなかなか楽しいですよー。キーボードにも対応してますから，PCゲームのようなノリでもプレイできますね！

竹崎　発売は7月26日。マウス同梱で5800円というふとっぱら価格だ！

奈美　…同梱，ということは「アウトトリガー」にもれなくマウスがついてくるってことですかー!? しかも，5800円!? えっと，えとえと，マウスは確か単品で2500円だからー，5800－2500＝3300円…。

竹崎　そう，とにかくお得ってことなんだ！

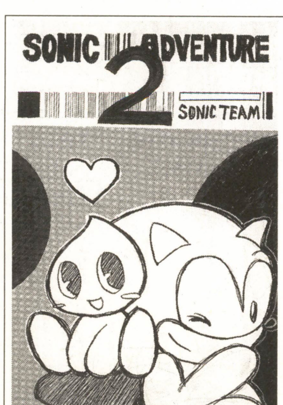
奈美　新キャラ2人の追加に，DCオリジナルモードも搭載! キャラエディット機能もついてめちゃくちゃ気合入ってまーす。

竹崎　皆，買ってね～。

奈美　以上，セガ竹崎と，らぶりーなっちがお送りしました。またね～!



◀せゆ　バーチャ4。個人的にも楽しみ♪ サラ使いのらぶりーなっちでした（笑）



◀ぱんだばば　お隣のくにきちさんと合作!?…と思いきや，違いました（笑）。まるで打ち合わせしたかのようです～



◀くにきち　どもです♪ HPいつも拝見してます。お仕事頑張ってください！



◀大山修史　あれっ，なんだかエリカ，騙されているような…。「エリカ」の「ラ」って一体…

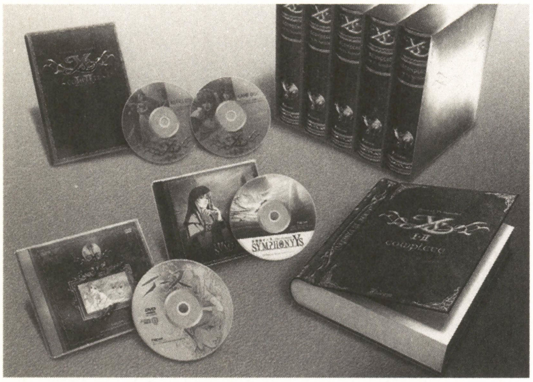
このコーナーを充実させるための，ご意見，お便り，イラストなどなどクビを長くしてお待ちしています。採用の方には特製テレカなど，非売品セガグッズをプレゼント。SEGA竹崎がお便りをすべて読んで，面白いコーナーを作っていきたいと思います。すべてのあてさきは，ここや!!
〒160-0032 東京都港区六本木1-4-30 六本木25森ビル18F （株）セガ カスタマーゲートウェイ部メディアコミュニケーションチーム「竹崎と奈美のベーセガ」係

セガCSパブリシティチームのHPはこちら→http://www.sega.co.jp/dreamcast/cspr/

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714
スタッフ募集中
採用担当BFN係
http://www.falcom.co.jp/

「イースⅠ・Ⅱ完全版」特典版は初回製造分のみ！



もうすでに幻（？）の豪華3大特典！
①復刻版OVAイース全7巻（DVD-VIDEO）＊3時間30分！
②交響曲イース21stセンチュリー（完全新作，新アレンジ）
③ゴールドケース入り「イースの書」パッケージ

遂に発売された「イースⅠ・Ⅱ完全版」ですが，今回も予約だけで完売したお店もあるようです。特典も今まで以上に豪華な内容でしたが，特典付きパッケージは，初回製造分のみとなってしまいました。したがって，現在店頭にある分のみで終了となりますので，まだ購入されていない方は，お急ぎを！

今回の「イースⅠ・Ⅱ完全版」は特典のほうに目がいきがちですが，ゲームそのものも，完全版の名にふさわしく，いろんな変更や改良がされています。その中でも，上級者用のナイトメアモードは，開発スタッフでもクリアしたのは，ほんの数人しかおらず，本当にクリアできるの？というくらいの激ムズモードです。腕に覚えのある方はぜひチャレンジしてみてね。それから，ボス戦タイムアタックは個人的にもうれしい機能で，まだ学生だった頃，PC-88版でイース，イースⅡをプレイしていて，クリアした後にボスと戦いたくて，フロッピーに各ボス戦の直前のデータをセーブしてたのを思い出します（当時はフロッピー１枚に１か所しかセーブできず，ボンビーな学生だったのでけっこう痛かった）。それが，もう今回は思う存分ボスと戦えるから，これだけでも買う価値ありだと思いますよ。

ヴァンテージマスターオンライン好評発売中！

ヴァンテージ・マスター・オンライン Vantage Master ONLINE

通信対戦機能が付いた「ヴァンテージマスターオンライン」がベクタープロレジから，好評発売中！インターネット接続，LAN接続，モデム接続，シリアルケーブル接続の4方式で接続でき，遠くにいる友達とも対戦できるぞ！コンピュータとの対戦も面白いけど，ヴァンテージマスターの醍醐味はなんと言っても，人対人の微妙なかけひきが楽しめるネットワーク対戦！隠しマスターだったアドルやジュリオ＆クリスなどファルコム歴代キャラも最初から選べるので，対戦もヒートアップすること間違いなし！
http://www.vector.co.jp/swreg/

現在ベクターで発売しているファルコム製品が一目でわかるこちらのコーナーもぜひご覧ください。ベクターでしか入手できないアイテムもあるぞ！
http://www.falcom.co.jp/vector/index.html

ツイムネットに対戦ロビー開設！

なかなか対戦相手が見つからない，全国の見知らぬ強豪と手合わせしたいという方に朗報！ツイムネットで，ヴァンテージマスターオンラインの対戦ロビーが開設されたので，対戦相手を探すのもこれで楽々！日本だけでなく韓国や台湾の強敵とも戦えるかも？さらに秋には，皆さんが心待ち？しているあの機能も予定されているらしいのでお楽しみ！今回のヴァンテージマスターオンライン発売のために，動作チェックを兼ねてプレイしたけど，やっぱりハマるハマる。たまに参加しているので，その時は勝負！
http://www.twimnet.co.jp/

パッケージ版「ヴァンテージマスターＶ２」との対戦も可能！すでにV2を持っている方は，友達にVMオンラインを薦めて熱い戦いを繰り広げよう！

ツヴァイ!!最新情報！



▲魔法を使っているシーンだ。
とにかく派手！アイテムの組み合わせによって魔法が変化するらしいので，アイテムを集めまくろう！

発売が待ち遠しいファルコム完全新作ツヴァイ!!現在完成に向けてラストスパートといったところです。ツヴァイの最新情報はファルコムホームページのHOT INFORMATIONで公開中なので要チェック！

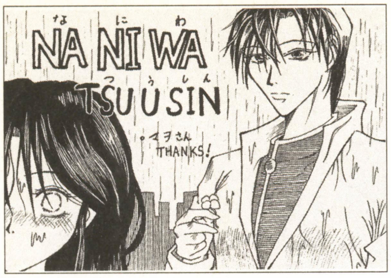
衝撃スクープ！こ，このイラストは！？



どっからどう見ても和風ですね～。これが次の次の新作？とすると，これまでのファルコム路線とはガラリと変わった作品になる可能性大！？このイラストを含め他の怪しげなイラストをファルコムホームページで公開中ですのでぜひご覧ください。
http://www.falcom.co.jp/info/



ちょっと話題が古くなりましたが，祝！日本代表コンフェデ杯準優勝！これでますます来年のワールドカップが楽しみになりましたね。この号が出る頃には，チケットGet！となってればいいんですけど・・。毎日ドキドキしながら郵便受けを見ている今日この頃です。



タイトル冠文：大阪府御堂筋太郎
タイトル・イラスト：千葉県・鈴木有人
本文：暁

夏バテ防止に なにわ通信

PRESENTED BY マイコンソフト ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F 電波新聞社大阪本社内
マイコンソフト株式会社　なにわ通信係
http://www.micomsoft.co.jp

この度，マイコンソフト社のホームページアドレスが変わることになりました。

新しいホームページ・アドレスは…，http://www.micomsoft.co.jp です。

今後とも，よろしくお願いいたします。

コンテンツ内容も少しずつ充実を図っていく予定です。

もし，弊社のホームページにリンクされている方がいらっしゃいましたら，お手数ですが，リンク先のアドレス変更をお願いいたします。

新人君入社

この度，マイコンソフト社に新人が入社しました。

彼には，ネットワーク関係の仕事全般をやってもらっています。今回のホームページ移転も，彼がWebサーバを用意してくれたおかげです。

これから，良い商品をたくさん作ってくれることを期待しています。

読者のプレゼント

投稿者の猿ともこさんが作られた同人誌を読者の皆さんにプレゼントさせていただきます。 執筆者は猿ともこさんをはじめ，ベーマガ読者の方々です。

同人誌は2冊なので，抽選になってしまうと思いますが，ドシドシご応募ください。

応募方法はハガキで「なにわ通信・猿ともこオールドゲーマーズ・ブック・プレゼント係」まで。

読者のプレゼント



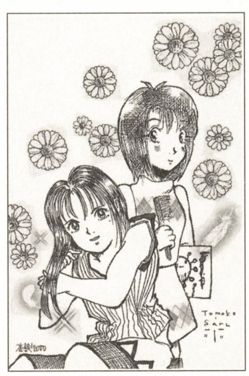
ありがたいおハガキのコーナー



大阪府・Aす
お久しぶりです。ちょっと昔のイラストですね。春の妖精さんでしょうか。とても可愛らしい～。



神奈川県・松田ばんれい
ちょっと懐かしいイラストです。なかなか美形のお兄さんですね。大学は今年卒業ですか？お勉強頑張ってください。



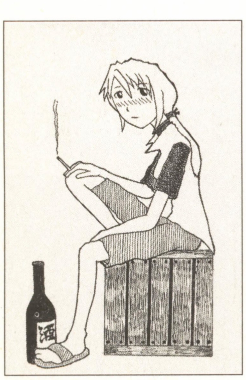
三重県・猿ともこ&福岡県・茶鈴
掲載が遅れて申し訳ありませんでした。昔のベーマガ・サークル仲間との合作。古くからの友人は大切にしたいものですね。



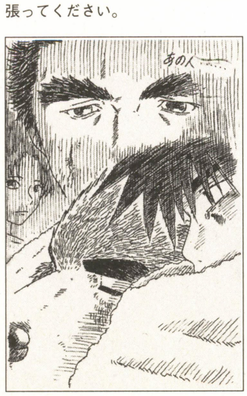
大阪府・藤井朱丸
ドイツ語の勉強頑張ってください。今年の冬は，アメリカへ留学されるそうですね。いっぱい勉強してきてくださいね。



宮城県・吏久
お忍びでどこかへお出かけでしょうか。ずっと描いていれば，必ずうまく描けるようになります。あきらめないことです。



千葉県・バグニンゲン
日本酒にタバコとは，ずいぶんと粋なお姉さんですねぇ。飲みすぎと，吸いすぎにはご注意ください。



大阪府・ARAI
彼の過去が少しずつ見えてきました。はたして熊さんは，彼の心を癒すことができるのか…。



北海道・青歌
どんなことでもそうですが，すぐにできる人はいません。絵もプログラムもマイペースで勉強すればいいと思いますよ。



愛知県・邪道坊や
今回は戦闘機乗りの2人を描いてくれました。先輩と後輩なのだそうです。先輩からの激励を受ける後輩君の様子が初々しい。



大阪府・桐谷欄様江
おしゃれな街角ですね。マヲさんの描かれるイラストの世界観は独特ですね。絵描きの仕事に就けばいいのに。いやマジで。

◆ 定期購読●直接購読のしおり ◆

弊社刊行の雑誌は，ご自宅直送方式の年間定期購読をお勧め申し上げます。スピーディーで確実，そしてお得な定期購読をぜひご利用下さいませ。
また　ご希望の雑誌・書籍が，ご近所の書店にない場合は右の振替用紙にて直接ご注文下さい。

■ 年間定期購読料（消費税込料金となっております。）

電　気　店	￥16,000	マイコンBASICマガジン	￥6,120
電波新聞縮刷版（CD-ROM版）	￥96, 000		

■ 年間購読の特典

①特別号も通常の定価でご購読いただけます。
②電気店は13冊分の購読ができます。

◆確実な振替口座を利用してご送金下さい◆

送金手数料が不要で，確実な郵便振替でご送金をお願い申しあげます。手続きは簡単です。右の郵便振替用紙に下記の必要事項を記入し，郵便局窓口に必要金額と共にご提出いただくだけですみます。
①購読申込書に必要事項を記入ください。なお１年間，或は２年間購読かをはっきりご記入ください。
②払込通知票，払込票に送金合計金額と住所，氏名をご記入ください。払込通知票と払込票は絶対に切りはなさないようにしてください。なお，必ずフリガナをご記入下さい。
③弊社への御意見・御希望がございましたら，通信欄へご記入下さい。
④すべて記入が終了しましたら，案内文のみ切り離して郵便局窓口に金額を添えて提出してください。

**∴振替によるご送金は，着本までに約２週間ほどかかりますのでお含みおきください。
お急ぎの場合は，現金書留に，右購読申込書を同封してご送金ください。**

この受領証は、郵便局で機械処理をした場合は郵便振替の払込みの証拠となるものですから大切に保存してください。

ご注意
この払込書は、機械で処理しますので、口座番号及び金額を記入する際は、枠内に丁寧に記入してください。
また、下部の欄（表面及び裏面）を汚したり、本票を折り曲げたりしないでください。

（郵政事業庁）

払込取扱票

02 東京

通常払込料金加入者負担

口座番号（右詰めにご記入ください）	金額
00150-3 51961	千 百 十 万 千 百 十 円 ※

加入者名：株式会社 電波新聞社

料金　特殊取扱

各票の※印欄は、払込人において記載してください。

通信欄

※定期購読誌名	購読期間	送金額（送料当社負担）
※新規 継続	年 月号より 年間	円

御希望誌・ソフト・グッズ名	冊数	送金額（送料共）
		円
		円

フリガナ 送付先住所 □□□-□□□□	産業分類	所属又は職種
フリガナ 送付先御氏名	電話 TEL	年令 才 / 性別 男・女

払込人住所氏名（郵便番号　　　）※

（電話番号　　－　　－　　）

受付局日附印

裏面の注意事項をお読みください。郵政事業庁（私製承認 東 第1005号）

切り取らないで郵便局にお出しください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

払込票兼受領証

口座番号：00150-3 通常払込料金加入者負担

右詰めにご記入ください 51961

加入者名：株式会社 電波新聞社

金額：千 百 十 万 千 百 十 円 ※

払込人住所氏名 ※

料金（消費税込み）円

受付局日附印

特殊取扱

切り取って郵便局へお出し下さい。

電波新聞社出版物一覧

（消費税込料金となっております）

2001.7 現

最新エレクトロニクス用語事典	5,300円	〒380円	DOS/V制作スペシャルNo.2～11	1,490円	〒340
半導体・ICのすべて	2,520円	〒310円	F-BASIC V6.3プログラミングのすべて	2,835円	〒310
ビデオ技術のすべて	2,100円	〒310円	ネットワーク対応データベース構築ガイド	2,854円	〒310
TVのすべて改訂版	1,988円	〒310円	パソコン認定試験合格対策試験問題集3・4級	2,100円	〒310
CD技術のすべて	2,039円	〒310円	デジタル1種〈技術科目〉最新テキスト	2,205円	〒340
MDのすべて	1,400円	〒310円	デジタル3種受験マニュアル	3,000円	〒340
センサーのすべて	2,000円	〒310円	アナログ・デジタル全資格〈法規科目〉	1,995円	〒340
改訂版DVD&DVC入門基本18章	1,529円	〒310円	シスアド受験スーパー記憶術	2,900円	〒340
増補改訂版デジタル放送がわかる本	1,600円	〒310円	基本情報受験スーパー記憶術	2,900円	〒340
増補改訂版・新ビデオ技術ハンドブック	2,500円	〒310円	特殊無線技士受験マニュアル・改訂新版	2,905円	〒340
新・まんが電気学入門	1,500円	〒310円	デジタルビデオカメラを使いこなす本シリーズ ビクター・ソニー・パナソニック・シャープ・キャノン	各 1,490円	各 〒310
新電気学入門講座	2,050円	〒310円			
新オシロ・ロジアナ入門講座	2,243円	〒310円	オールアバウト カプコン対戦格闘ゲーム1987～2000	2,100円	〒450
電子計測器ガイドブック	7,500円	〒450円	オールアバウト SNK対戦格闘ゲーム 1991～2000	2,310円	〒450
IT社会のビジネスプラットホーム	2,625円	〒310円	IT革命にどう生き残るか	2,205円	〒310

※上記以外にも新聞・出版物がございますので最寄りの本社、支局へお問い合わせ下さい。

東京本社 03-3445-8201(ダイヤルイン)
大阪本社 06-6203-3361
西部本社 092-431-7411
札幌支局 011-641-5591
仙台支局 022-227-7211
新潟支局 025-245-2526
関東総局 027-326-3206
松本支局 0263-36-2266
名古屋支局 052-261-4541
金沢支局 076-263-8661
京都支局 075-221-8021
広島支局 082-228-5581
高松支局 087-861-3111
鹿児島支局 099-226-3630

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　Q10で④の方のみ，どちらを購入する予定ですか？
①携帯 ②PHS ③両方 ④未定

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯		応募年月日		年　月　日

キリトリ線

郵便はがき

50円切手をお貼り下さい

141-8715

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン 編集部

8月号 行

（フリガナ）		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　］

ベーマガお楽しみプレゼント⑧ 応募欄（78ページ欄外）	希望番号	品名
	番	

キリトリ線

月刊マイコンBASICマガジン⑧ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想(本誌への掲載／可・不可)

ペンネーム：

■アンケート回答記入欄 ■ ※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1	□	Q2	□ 〔メーカー名 〕
Q3	□	Q4 □	Q5 □
Q6	□□□□□□□□		
Q7	□□□□□	〔商品名 〕	
Q8	□□□□□□□□		
Q9	□□□□□□□□		
Q10	□	Q11 □	
Q12	〔 〕	Q13 〔 〕	
Q14	□□□	Q15 □□□	

キリトリ線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Macintosh ⑦UNIX ⑧Linux ⑨その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームボーイ ⑥ゲームボーイカラー ⑦ゲームボーイアドバンス ⑧ワンダースワン ⑨ワンダースワンカラー ⑩その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など


イース
I・II
コンプリート
完全版
PCゲーム
定番
“本当の冒険”への回帰…。分かたれた物語は、今、ここに一つの神話となって甦る。
■画面に映る全てが、人の根底にある未知への憧れや冒険に胸躍る感覚を呼び覚まし、イースの物語の中へ導いてくれる冠絶のビジュアル！
■抜群の操作性に、絶妙なバランスと緊張感。プレイヤー自身が成長していくような感覚こそが、A・RPGであるイースの特徴であり醍醐味！
■物語を楽しみたい方から熾烈な戦闘を極めたい方まで、幅広いユーザーに対応した4つのモードが選択可能。ボス戦だけの「タイムアタック」も追加！
■全体の統一感を高めさらなる飛躍を迎えたイースサウンド。さらに効果音の臨場感を増すなど音の演出的な面も一層の強化！
Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/
「イースI・II完全版」についての情報は、ホームページをご覧下さい。http://www.falcom.com/
■ご購入は便利なオンラインショッピングコーナーをご利用下さい falcom.com(ファルコムドットコム)
■社員募集(新卒・既卒・中途)！詳しくはファルコムホームページ「会社案内・社員募集」をご覧ください。
7/19発売
WindowsMe/98/95
(Windows2000対応)

定価550円 本体524円 雑誌18361-08 T1118361080551

第二十巻第八号通巻二三〇号 昭和五十七年十二月六日第三種郵便物認可 二〇〇一年七月二十五日印刷 二〇〇一年八月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一－十一－十五 電話〇三(三四四五)六一一一

2001

わかる！動かせる！
プログラムが組める雑誌

マイコンBASIC Magazine

8

プレステ2でゲームをプログラム！

http://www.basicmagazine.net/

電波新聞社